

GAMES

COMMODORE 64

COMPUTERS PLAY



Robert Young Roger Bush

**GAMES
COMMODORE 64
COMPUTERS
PLAY**

GAMES COMMODORE 64 COMPUTERS PLAY

BY

**ROBERT YOUNG
ROGER BUSH**



ADDISON-WESLEY PUBLISHING COMPANY

Sydney · London · Reading, Massachusetts · Menlo Park, California ·
Amsterdam · Don Mills, Ontario · Manila · Singapore · Tokyo

© 1984 ADDISON-WESLEY PUBLISHERS LIMITED

All rights reserved. No part of this publication may be reproduced, stored in retrieval system, or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission of the publisher.

The programs presented in this book have been included for their instructional value. They have been tested with care but are not guaranteed for any particular purpose. While every care has been taken, the publishers cannot be held responsible for running mistakes which may occur.

ISBN 0-201-14428-X

ABCDEF 987654

The Authors:

ROBERT YOUNG is a veteran of the computer book field, having written or co-written nine books which have been published in the last 18 months. His works range from *The Giant Book of Dragon Games* (with Roger Bush and Alan Blackman, Fontana, 1984), *Creating Adventures on your Commodore 64* (Interface Publications, 1984) and *How to Program the Commodore 64* (Ballantine, 1984). Formerly a professional photographer, Robert is currently working – with Roger Bush – on a major book on wargaming, with and without the computer. In his early thirties, Robert lives with his wife and son in Geelong, a city about 40 miles from Melbourne. His leisure activities include reading and writing science fiction and working on an early model sports car he owns.

ROGER BUSH, in his mid-twenties, has collaborated with Robert Young on a number of books including *Dynamic Games for your Dragon* (Interface Publications, 1983), and *The Giant Book of VIC 20 Games* (Fontana, 1984). He works in the employment department of the Australian Public Service, and spends the majority of his leisure time on his passion of war-gaming. He has run out of opponents in his local area; no-one wants to play with him anymore, as defeat at his hands is almost inevitable. Roger is using this knowledge to write, with Robert Young, a major work on war-gaming. Roger has a vast and enviable collection of historically accurate miniature soldiers.

CONTENTS

Foreword — Tim Hartnell.....	ix
Introduction	1

ARCADE ACTION

Leaper.....	7
Joystick version	9
Keyboard version.....	13
Descent into Chaos.....	18
Joystick version	19
Keyboard version.....	22
Fire	25
Skiing.....	32
Joystick version.....	34
Keyboard version	38
Spotlight	43
Astro-Blaster.....	50
Cosmic Journey.....	57
Space Race.....	61
Ghostie	65
Joystick version.....	67
Keyboard version	71
Astrosplash	75
Joystick version.....	77
Keyboard version	82

ADVENTURE AND SIMULATION

Survival	91
----------------	----

Roman Empire.....	102
Battle of Britain.....	115
Sunnybrook Farm.....	126

CHALLENGE THE COMPUTER

3-D Noughts and Crosses	145
Statesman.....	156
Unscramble Alien.....	159
Reversi	168
Chessboard Nim.....	173

APPENDIX

Writing your own games.....	177
-----------------------------	-----

FOREWORD

I first met Robert Young more than ten years ago, when we were both working for (and I shudder to think of it now) the Public Service arm of the Victoria Police Department. I remember we spent a lot of time plotting (but unfortunately never putting into action) ways of inserting bogus 'crimes' into the department's massive mainframe computer.

There was no way – in those days – for ordinary mortals to possess computers, so our computer game-playing had to be done in imagination. After I left the department, and started plying various trades around the world, Rob became addicted to computer game-playing after a chance encounter with the massive adventure program ZORK on a university time-sharing system. In London, I bought a ZX80, the first machine I could actually afford, and a year later, in Geelong, Robert Young bought his first computer, a TRS-80.

With Roger Bush, Rob formed a company designed to market computer games. Their first product, a war game, was a significant failure. Significant in that the production and advertising costs wiped out all their meagre capital, and the company faded away.

I'd kept in touch with Rob over the years, and on one of my trips to Australia at the end of 1982, called in to see him one day and noticed he'd recently bought a Tandy Color Computer. We started talking about the possibilities of computer books. It was obvious that Roger would be involved, due to his computer interest.

It has proved a particularly fruitful relationship for all of us. Since that time, Rob (sometimes on his own, and in other cases in collaboration with Roger or myself) has written some nine books on computers and computer gaming which have been published around the world.

This is his latest, and – without doubt – the strongest book to date. The two authors commissioned a number of programs from some of Australia's most brilliant young programmers,

and wrote the others themselves, to produce a balanced collection of very worthwhile programs for your Commodore 64.

Whether you're looking for arcade games, the action and excitement of Adventure and simulation programs, or for a series of challenges to your brainpower, you'll find the programs in this collection. And because they were created specifically for the Commodore 64, they make the maximum use of the 64's powerful colour and sound capabilities.

It's time you started a-gaming. After all, the ROMAN EMPIRE, the dizzy planet of CHAOS and the criminals on the prison planet of URIEL are waiting for you to bring meaning and order into their lives.

Grasp your joystick firmly in your hand, and plunge into the world of Games Commodore 64's Play.

Tim Hartnell,
London, 1984

Tim Hartnell is the author of more than 30 books on microcomputers. Recent works include GAMES BBC COMPUTERS PLAY (Addison-Wesley/Interface, 1982), THE GIANT BOOK OF COMPUTER GAMES (Fontana, UK, 1983 and Ballantine, USA, 1984) and PROGRAMMING YOUR ZX SPECTRUM (Interface, 1982). He is currently involved in writing a major book on the Sinclair QL computer.

INTRODUCTION

In this book you will find an exciting collection of games for your Commodore 64.

Arcade action can be found in games such as GHOSTIE and ASTROSMASH, which use the sprite graphics and sound generating capabilities of your Commodore 64 to the full.

LEAPER will provide you with many hours of fun. This game features programmable characters, great sound and nine levels of difficulty.

We have also included several simulation games. You will have the opportunity to command the 11th Fighter Group, Royal Air Force, during a critical two-week period in September, 1940. Can you win the Battle of Britain?

Your management skills will be tested to the full as you attempt to run SUNNYBROOK FARM in the face of a harsh economic climate.

We have had a great deal of enjoyment putting together this collection of games and hope that you will gain as much pleasure from playing them.

When typing in the programs, read each line carefully and make sure that you enter it exactly. It is always a good idea to SAVE a copy of the program as soon as it is finished and before you RUN it. This is especially so in the case of moving graphics games where a typing error can cause the system to crash or the keyboard to freeze, thereby losing the program.

After typing in a program, a debugging session may be required to track down any errors which may have occurred during the typing stage. Running the program and fixing the errors as they appear should be sufficient, but if the program doesn't seem to be running correctly, it may be necessary to go through it line by line, comparing it to the original program in the book. A simple error, such as leaving out a semi-colon or leaving a NEXT off the end of a FOR/NEXT loop can cause untold havoc in a program!

Where possible we have included REM statements to explain control characters, graphic characters and any other items which may cause confusion in the listings. Unless you particularly wish to leave them in, it is best to drop them from your version of the game. We have also spaced out the statements and command words to make each line easier to read. You may also find it better to leave out the spaces and close the lines up. These alterations will make the programs run more efficiently.

Following this introduction is a full list of the control characters and graphics characters along with the symbols which appear in the program listings.

Robert Young, Roger Bush
Geelong, Australia, 1984.

CONTROL CHARACTERS

CLEAR SCREEN	=	␣	BLUE	=	␣
HOME CURSOR	=	␣	YELLOW	=	␣
DOWN CURSOR	=	␣	ORANGE	=	␣
UP CURSOR	=	␣	BROWN	=	␣
RIGHT CURSOR	=	␣	LT RED	=	␣
LEFT CURSOR	=	␣	GRAY 1	=	␣
BLACK	=	␣	GRAY 2	=	␣
WHITE	=	␣	LT GREEN	=	␣
RED	=	␣	LT BLUE	=	␣
CYAN	=	␣	GRAY 3	=	␣
PURPLE	=	␣	REV ON	=	␣
GREEN	=	␣	REV OFF	=	␣

GRAPHICS CHARACTERS

USING SHIFT KEY

+ KEY = + H KEY = |

- KEY = | J KEY = ^

E KEY = ▯ K KEY = /

O KEY = # L KEY = L

W KEY = O Z KEY = #

E KEY = ~ X KEY = #

R KEY = _ C KEY = ~

T KEY = | V KEY = X

Y KEY = | B KEY = |

U KEY = / N KEY = /

I KEY = ^ M KEY = \

O KEY = Γ

P KEY = 7

@ KEY = 1

* KEY = ~

A KEY = #

S KEY = #

D KEY = ~

F KEY = _

G KEY = |

USING COMMODORE KEY

+ KEY = ☼ J KEY = ⑆

- KEY = ☿ K KEY = ⑇

£ KEY = ☿ L KEY = ⑈

Q KEY = † Z KEY = ‡

W KEY = ‡ X KEY = †

E KEY = † C KEY = ⑈

R KEY = ‡ V KEY = ⑈

T KEY = ‡ B KEY = ⑈

Y KEY = ‡ N KEY = ⑈

U KEY = ‡ M KEY = ⑈

I KEY = ⑈

O KEY = ⑈

P KEY = ⑈

@ KEY = ⑈

* KEY = ⑈

A KEY = ⑈

S KEY = ⑈

D KEY = ⑈

F KEY = ⑈

G KEY = ⑈

H KEY = ⑈

ARCADE ACTION

Leaper

Descent into Chaos

Fire

Skiing

Spotlight

Astro-Blaster

Cosmic Journey

Space Race

Ghostie

Astrosmash



ghostie

LEAPER

Leaper is an exciting game which uses programmable characters to produce a fast-moving graphics game which requires a high level of skill to play successfully.

The object of the game is to control a bouncing man and break 21 balloons in the shortest possible time. The man jumps up from platforms near the bottom of the screen and can be made to jump straight up, left or right and he can be moved slightly to either side while he is in the air.

There are two listings for this game. The first one is for a joystick-controlled game in which the man is moved left and right. If you don't use the joystick to bounce the man straight up in the air from time to time, he will bounce lower and lower until he eventually stops. The second version of the game is for keyboard operation. In this version the "A" key and the "D" key move the man left and right respectively and the "W" key makes him leap straight up.

There are nine levels of difficulty in Leaper. On each level the platforms are placed in different positions and are of varying widths.

At the beginning of the game, you are asked which level of difficulty you wish to try. There is also an option for the computer to select a level at random. After entering the level you require, there is a short delay while the character set is defined. The screen will then clear and be set to black with several platforms and a number of balloons floating in the air. The leaping man enters from the left hand side of the screen and jumps off the top of the platform. The game is on and it is now up to you to control the leaper.

Lines 10 to 12 PRINT the title screen and ask for the level of difficulty. Line 13 sends the program to the routine in lines 500 to 570. This routine takes the character set out of the ROM and POKES in the programmable characters. Line 14 clears the screen and calls up the routine in lines 600 and 601 which initialise the sound values.

Lines 15 to 20 call up the required level of difficulty and PRINT the platform on the screen. You will see these statements as rows of the letter "D" in the program listings. They appear on the screen as a solid platform. Line 25 then PRINTs the balloons into randomly-selected locations. Line 26 then calls up the routine in line 250 to 280 which sends the leaping man out onto the screen.

Line 30 POKEs a blank space into the last location occupied by the man. This gives the effect of continuous movement and prevents more than one leaper from appearing on the screen at the same time.

The joystick (or keyboard) input is read by line 31. Line 32 calls up the sound routine which provides the 'beep'. This routine is in line 620 to 630.

Lines 35 and 36 control the left and right movement of the man while he is in the air. The two lines act on the input and add .1 to the value of X. This means that after 10 such value changes the man will alter his position across the screen (the X axis). Lines 40 to 44 control the vertical movement of the leaper. Line 45 checks to see if the man has hit the balloon. Line 50 POKEs the leaper onto the screen and line 52 sets the man's color. Value one sets him to white. You can make him any color you choose by altering this value.

Line 55 stops the leaper from falling off the screen and line 58 checks to see if he has missed a platform. His jumping height is checked and limited in line 60. Variable LMT is the limit to which he is allowed to jump.

Line 90 goes back to the start of the main program which begins at line 30.

The sound of the popping balloon is called up by the routine in lines 100 and 105. These lines also POKE the explosion on the screen and increment the variable HIT. The character used for the explosion is actually an asterisk from the normal character set. Used in this way it is very effective.

Lines 150 to 160 control the direction of the leaper. LD is the variable and value one is for left, value 2 for up and value 3 for right. These values are used to control the direction the man bounces in when he leaves the platform. The values of LD are set

in lines 210 to 220.

Lines 180 to 190 PRINT the score after the 21 balloons have been burst and the game is over.

The routine in lines 200 to 220 is called up when the man jumps up from the platform. It uses the variable LD to control the direction of the bounce. The starting routine is handled by lines 250 to 280 and lines 300 to 315 control the failure messages.

The platforms for the different levels of difficulty are contained in lines 400 to 480 and are put on the screen uses PRINT statements.

Leaper is a fun game to play and should give you hours of entertainment. The program was written by William Leader.

```
1 REM *** LEAPER JOYSTICK VERSION ***
5 S=54272:HF=60
10 POKE53281,7:POKE53280,7:PRINT"~~~~~
  ~~~~~WHAT LEVEL(0-9/0=RND)???"
11 GET A$:IF A$="" THEN 11:REM CLR SCRN DOWN
  CRSR BLUE RIGHT CRSR
12 R=VAL(A$):IF R<1 THEN R=INT(9*RND(1))+1
13 GOSUB 900:GOTO 500
14 PRINT"~~~~~":GOSUB 600:REM CLR SCRN
  DOWN CRSR
15 FOR T=1 TO 30:PRINT:NEXT
20 ON R GOTO 400,410,420,430,440,450,460,470,
  480,490
25 FOR L=1 TO 30:POKE 1024+INT(RND(1)*400),5:
  NEXT:POKE 53281,0:TI$="000000"
26 GOTO 250
30 POKE(1024+X+40*Y),32
31 JV=PEEK(56320):FR=JV AND 16:JV=15-(JV AND 15)
32 POKE S+4,32:GOSUB 620
35 IF JV=4 THEN X=X-.1
36 IF JV=8 THEN X=X+.1
40 IF D=-1 THEN 150
41 IF D=1 THEN Y=Y+1:CN=1
```

```

43 IF D=1 AND PEEK(1024+X+40*Y)=4 THEN 200
44 IF D=-1 AND PEEK(1024+X+40*Y)=4 THEN Y=Y+1
   :D=1:W=-1
45 IF PEEK(1024+X+40*Y)=5 THEN 100
50 POKE (1024+X+40*Y),CN
52 POKE 55296+X+40*Y,1
55 IF X<1 OR X>39 THEN D=1:W=-1
58 IF Y>23 THEN 300
60 IF Y<LMT THEN FOR T=1 TO 100:NEXT:D=1:W=-1
90 GOTO 30
100 POKE 1024+X+40*Y,42:POKE 55296+X+40*Y,10:
   HIT=HIT+1:IF HIT>20 THEN 180
105 GOSUB 650:GOTO 30
150 IF LD=1 THEN Y=Y-1:CN=1:GOTO 41
155 IF LD=2 THEN Y=Y-1:X=X-1:CN=3:GOTO 41
160 IF LD=3 THEN Y=Y-1:X=X+1:CN=2:GOTO 41
180 POKE 53281,1:POKE 1024+X+40*Y,CN
181 GOSUB 710:POKE 53281,14:POKE 53280,14
182 SC=VAL(TI$):POKE 53272,21:PRINT"XXXXXXXXXXXXX
   YOU SCORED      ":INT(SC)/100
183 REM CLR SCRN DOWN CRSR BLUE
185 FOR T=1 TO 2000:NEXT:PRINT"XXXXXXXX DO YOU WANT
   ANOTHER GAME?"
190 GET A$:IF A$<>"Y" THEN 195
192 RESTORE:RUN
195 IF A$<>"N" THEN 190
197 END
200 POKE 1024+X+40*(Y-1),0:Y=Y-1
201 FOR T=1 TO 5:W=1
205 JV=15-(PEEK(56320) AND 15)
210 IF JV=1 THEN LD=1:LMT=LMT-1.5:IF LMT<1 THEN
   LMT=0:GOTO 220
211 IF JV=4 THEN LD=2:GOTO 220
212 IF JV=8 THEN LD=3:GOTO 220
215 NEXT
217 LD=1:LMT=LMT+.5:IF LMT>21 THEN LMT=21
220 D=-1:LMT=LMT+1:GOTO 30
250 PRINT"XXXXXXXXXXXXX":REM HOME DOWN CRSR
251 FOR T=1 TO 1000:NEXT
255 FOR T=0 TO 5:POKE1024+T+40*4,0:FOR I=1 TO
   100:NEXT:POKE1024+T+40*4,32:NEXT

```

```

256 POKE 1024+5+40*4,0:FOR I=1 TO 1000:NEXT
260 FORT=4T00STEP-1:POKE1024+5+40*T,1:FOR I=1TO
    100:NEXT:POKE1024+5+40*T,32:NEXT
261 FOR I=1 TO 100:NEXT
265 FOR T=0 TO 4:POKE1024+5+40*T,1:FOR I=1 TO
    100:NEXT:POKE1024+5+40*T,32:NEXT
270 POKE1024+5+40*4,0:FOR T=1 TO 1000:NEXT:X=5:
    Y=4:CN=1:POKE1024+5+40*4,32:D=-1
275 POKE 1024+X+40*Y,1:LD=3
277 PRINT"XXXXXXXXXX":REM HOME CRSR DOWN
280 GOTO 30
300 POKE 1024+X+40*Y,6:GOSUB 670
310 POKE53281,14:POKE53272,21:PRINT"XXXXXXXXXXYOU
    FAILED AFTER ONLY GETTING ";
312 REM CLR SCRN BLUE DOWN CRSR
315 PRINT HIT;"          BALLOONS":GOTO
    185
400 PRINT"#####
    DDD":GOTO 25:REM LT GREEN
410 PRINT"#####          #####
    DDD":GOTO25
420 PRINT"#####XXXXXXXXXX#####
    #####TTTTTTTT#####":GOTO 25
425 REM USE ABBREVIATION FOR PRINT OR LINE
    WON'T FIT "J"=UP CRSR
430 PRINT"#####TTTTTTTT#####
    XXXXXXXXXX#####":GOTO25
440 PRINT"##### DDDD DDDD DDDD DDDD DDD
    DDD ":GOTO25
450 PRINT"#####TTTTTTTT#####XXXXXXXXXX#####
    #####TTTTTT";
451 PRINT"#####TTTTTTTT":GOTO25
460 PRINT"#####TTTTTTTTTTTTTTTTTTTT
    #####XXXXXXXXXX":
461 PRINT"#####":GOTO25
470 PRINT "#####
    DDD";
471 PRINT"XXXXXXXXXX"

```

12


```

700 RETURN
710 FOR L=0 TO 24:POKE S+L,0:NEXT
720 POKE S+3,9:POKE S+5,41:POKE S+6,89:POKE S+14,
    117:POKE S+18,16:POKE S+24,143
725 READ FR,DR:IF FR=0 THEN RETURN
730 POKES+4,65:FORT=1TODR#2:FQ=FR+PEEK(S+27)/2:
    HF=INT(FQ/256):LF=FQAND255
740 POKE S+0,LF:POKE S+1,HF:NEXT:POKE S+4,64:
    GOTO 725
800 DATA 4817,2,5103,2,5407,2,8583,4,5407,2,
    8583,4
805 DATA 5407,4,8583,12,9634,2,10207,2,10814,2,
    8583,2
810 DATA 9634,4,10814,2,8583,2,9634,4,8583,12,
    0,0
900 PRINT"#####ENTER LEVEL#
    ":R:RETURN
910 REM CLR SCRN DOWN CRSR RIGHT CRSR RED

```

```

1 REM *** LEAPER KEYBOARD VERSION ***
5 S=54272:HF=60
10 POKE53281,7:POKE53280,7:PRINT"#####
    #####WHAT LEVEL(0-9/0=RND)???"
11 GET A$:IF A$="" THEN 11:REM CLR SCRN DOWN
    CRSR BLUE RIGHT CRSR
12 R=VAL(A$):IF R<1 THEN R=INT(9*RND(1))+1
13 GOSUB 900:GOTO 500
14 PRINT"#####":GOSUB 600:REM CLR SCRN
    DOWN CRSR
15 FOR T=1 TO 30:PRINT:NEXT
20 ON R GOTO 400,410,420,430,440,450,460,470,
    480,490
25 FOR L=1 TO 30:POKE 1024+INT(RND(1)*400),5:
    NEXT:POKE 53281,0:TI$="000000"

```

```

26 GOTO 250
30 POKE(1024+X+40*Y),32
31 JV=PEEK(197):FR=16:IF JV=64 THEN FR=10
32 POKE S+4,32:GOSUB 620
35 IF JV=10 THEN X=X-.1
36 IF JV=18 THEN X=X+.1
40 IF D=-1 THEN 150
41 IF D=1 THEN Y=Y+1:CN=1
43 IF D=1 AND PEEK(1024+X+40*Y)=4 THEN 200
44 IF D=-1 AND PEEK(1024+X+40*Y)=4 THEN Y=Y+1:
    D=1:W=-1
45 IF PEEK(1024+X+40*Y)=5 THEN 100
50 POKE (1024+X+40*Y),CN
52 POKE 55296+X+40*Y,1
55 IF X<1 OR X>39 THEN D=1:W=-1
58 IF Y>23 THEN 300
60 IF Y<LMT THEN FOR T=1 TO 100:NEXT:D=1:W=-1
90 GOTO 30
100 POKE 1024+X+40*Y,42:POKE 55296+X+40*Y,10:
    HIT=HIT+1:IF HIT>20 THEN 180
105 GOSUB 650:GOTO 30
150 IF LD=1 THEN Y=Y-1:CN=1:GOTO 41
155 IF LD=2 THEN Y=Y-1:X=X-1:CN=3:GOTO 41
160 IF LD=3 THEN Y=Y-1:X=X+1:CN=2:GOTO 41
180 POKE 53281,1:POKE 1024+X+40*Y,CN
181 GOSUB 710:POKE 53281,14:POKE 53280,14
182 SC=VAL(TI$):POKE 53272,21:PRINT"*****
YOU SCORED      ";INT(SC)/100
183 REM CLR SCRN DOWN CRSR BLUE
185 FOR T=1 TO 2000:NEXT:PRINT"*****DO YOU WANT
    ANOTHER GAME?"
190 GET A$:IF A#<>"Y" THEN 195
192 RESTORE:RUN
195 IF A#<>"N" THEN 190
197 END
200 POKE 1024+X+40*(Y-1),0:Y=Y-1
201 FOR T=1 TO 5:W=1
205 JV=PEEK(197)
210 IF JV=9 THEN LD=1:LMT=LMT-1.5:IF LMT<1 THEN
    LMT=0:GOTO 220
211 IF JV=10 THEN LD=2:GOTO 220
212 IF JV=18 THEN LD=3:GOTO 220

```

```

215 NEXT
217 LD=1:LMT=LMT+.5:IF LMT>21 THEN LMT=21
220 D=-1:LMT=LMT+1:GOTO 30
250 PRINT"XXXXXXXXXXXX":REM HOME DOWN CRSR
251 FOR T=1 TO 1000:NEXT
255 FOR T=0 TO 5:POKE1024+T+40*4,0:FOR I=1 TO
    100:NEXT:POKE1024+T+40*4,32:NEXT
256 POKE 1024+5+40*4,0:FOR I=1 TO 1000:NEXT
260 FORT=4TDOSTEP-1:POKE1024+5+40*T,1:FOR I=1 TO
    100:NEXT:POKE1024+5+40*T,32:NEXT
261 FOR I=1 TO 100:NEXT
265 FOR T=0 TO 4:POKE1024+5+40*T,1:FOR I=1 TO
    100:NEXT:POKE1024+5+40*T,32:NEXT
270 POKE1024+5+40*4,0:FOR T=1 TO 1000:NEXT:X=5:
    Y=4:CN=1:POKE1024+5+40*4,32:D=-1
275 POKE 1024+X+40*Y,1:LD=3
277 PRINT"XXXXXXXX":REM HOME CRSR DOWN
280 GOTO 30
300 POKE 1024+X+40*Y,6:GOSUB 670
310 POKE53281,14:POKE53272,21:PRINT"XXXXXXXXXXXX
    YOU FAILED AFTER ONLY GETTING ";
312 REM CLR SCRN BLUE DOWN CRSR
315 PRINT HIT:"          BALLOONS":
    GOTO 185
400 PRINT"#####
    DDD":GOTO 25:REM LT GREEN
410 PRINT"#####          #####
    DDD":GOTO25
420 PRINT"#####XXXXXXXXXXXX#####
    #####TTTTTTTT#####":GOTO 25
425 REM USE ABBREVIATION FOR PRINT OR LINE WON'T
    FIT "T"=UP CRSR
430 PRINT"#####TTTTTTTT#####
    XXXXXXXXXXXX":GOTO25
440 PRINT"##### DDDD DDDD DDDD DDDD DDD
    DDD ":GOTO25
450 PRINT"#####XXXXXXXXXXXX#####
    #####TTTTTTTT";
451 PRINT"#####TTTTTTTT":GOTO25

```

```

460 PRINT"#####TTTTTTTTTTTTTTTT#####
   #####";
461 PRINT"#####":GOTO25
470 PRINT "#####
   DDD";
471 PRINT"#####";
472 PRINT"TTTTTTTTTTTTTTTT#####
   #####";GOTO 25
480 PRINT"TTTTTTTTT #####
   #####":GOTO 25
500 POKE 56334,PEEK(56334)AND254:POKE1,PEEK(1)
   AND251:FOR I=0 TO 63:FOR J=0 TO 7
505 POKE 12288+I*8+J,PEEK(53248+I*8+J):NEXT:NEXT
   :POKE 1,PEEK(1) OR 4
510 POKE 56334,PEEK(56334) OR 1:POKE 53272,
   (PEEK(53272) AND 240)+12
512 FOR CH=0 TO 6:FOR BY=0 TO 7:READ A:POKE
   12288+(8*CH)+BY,A
515 NEXT:NEXT
550 DATA 24,24,60,102,153,24,24,60,153,90,60,36,
   24,36,66,129
555 DATA 12,12,240,44,58,72,132,129,96,96,62,104
   ,184,68,66,193
560 DATA 0,0,0,255,255,0,0,0,60,126,126,126,126,
   126,60,24
565 DATA 24,24,36,36,60,60,195,129
570 GOTO 14
600 FOR L=0 TO 24:POKE S,L:NEXT:POKE S+5,9:POKE
   S+6,0:POKE S+24,15:S0=100:W=1
601 RETURN
620 HF=HF+W:IF HF<1 THEN HF=1
625 POKE S+5,15:POKE S+6,54:POKE S+1,HF:POKE
   S+4,17
630 RETURN
650 POKES+4,32:POKES+5,15:POKES+1,40:POKES,100:
   POKES+4,129:FOR T=1 TO 30:NEXT
660 RETURN
670 FOR L=0 TO 24:POKE S+L,0:NEXT:POKE S+5,20

```

```

671 POKE S+24,15
675 FORT=20T080:POKES+4,32:POKES+1,T:POKES,T+20:
    POKES+4,33:FORI=1T010:NEXT:NEXT
680 FOR T=1 TO 50
685 IFT/15=INT(T/15)THENPOKES+4,32:POKES+1,20:
    POKES,50:POKES+4,33:POKES+5,15
690 NEXT
700 RETURN
710 FOR L=0 TO 24:POKE S+L,0:NEXT
720 POKE S+3,8:POKE S+5,41:POKE 6,89:POKE S+14,
    117:POKE S+18,16:POKE S+24,143
725 READ FR,DR:IF FR=0 THEN RETURN
730 POKES+4,65:FORT=1TODR*2:FQ=FR+PEEK(S+27)/2:
    HF=INT(FQ/256):LF=FQAND255
740 POKE S+0,LF:POKE S+1,HF:NEXT:POKE S+4,64:
    GOTO 725
800 DATA 4817,2,5103,2,5407,2,8583,4,5407,2,
    8583,4
805 DATA 5407,4,8583,12,9634,2,10207,2,10814,2,
    8583,2
810 DATA 9634,4,10814,2,8583,2,9634,4,8583,12,
    0,0
900 PRINT"*****ENTER LEVEL:
    ";R:RETURN
910 REM CLR SCRN DOWN CRSR RIGHT CRSR RED

```

DESCENT INTO CHAOS

Descent into Chaos is a moving graphics game in which you fly your space ship down a long winding tunnel on the planet Chaos. The game requires concentration and good reflexes to be completed successfully.

The first listing of the program is for joystick control. You move your ship left and right with the joystick, and slow your descent by pressing the fire button.

The second listing is for a keyboard-operated game. In this version, the “,” key and the “.” key move the space ship left and right. This should be easy to remember because the ‘less than’ and ‘greater than’ symbols on the top of these two keys point in the direction of travel. The “T” key can be used to slow your headlong rush.

After the title is displayed, the screen clears and is set to black. Your space ship, a diamond shape, appears at the top of the screen. The planet Chaos and the opening of the tunnel are printed and your space ship starts falling into the cavern. You must avoid hitting the sides while trying to get your ship all the way to the bottom of the tunnel. You have four “lives”. After they’ve been used up, you’ll be told just how deep into the tunnel you managed to go.

Beware, the tunnel gets narrower as it gets nearer to the centre of the planet.

The main part of the program lies between line 80 and the NEXT in line 180. This is the loop the program goes through as it prints the planet and tunnel.

Line 40 calls up the routine to initialise the sound generator and then provides the title display. Line 50 sets the screen to black and initialises the variable used to read the joystick (DD). Line 60 clears the screen and sets up the string S\$. This string consists of a number of blank spaces used to create the tunnel.

Line 70 initialises the value of Q (used to TAB S\$ across the screen), sets SC to 1024 (the start of the screen memory) and gives X the value of 14 (X is used to give your space ship its distance across the screen).

Line 80 starts off our game loop and prints 38 reversed spaces on the screen. These spaces appear as a blue bar. Line 85 checks to see how wide our tunnel is and, when it reaches two spaces, calls up the end routine. Lines 90 to 130 choose a random number and use it, along with variable Q, to TAB S\$ across the screen. This changing value causes the twists and turns in the tunnel. Lines 110 and 120 are safeguards to prevent the tunnel from running off the screen.

Line 140 uses SC and X to POKE your spaceship onto the screen. Line 150 calls up the routine in lines 5000 and 5010 which read the joystick input. This line then checks to see if the fire button has been pressed and, if it has, adds a short delay which slows down the game. Lines 155 and 157 act on the input by altering the value of X.

Line 170 checks to see if you have collided with the side of the tunnel. If you have, line 1000 increments variable H, which counts the number of collisions. Lines 1010 to 1030 provide the sound of the crash. Lines 1050 and 1060 check to see which side of the tunnel you have hit and move your ship back into the centre to continue the game.

Line 180 increments the variable M, which is used to keep a record of how far down the tunnel you manage to fly.

After the RR FOR/NEXT loop has run through, the rest of line 180 subtracts one space from the string S\$. This makes the tunnel narrower. We then go back to line 80 for another cycle.

```
10 REM *** DESCENT INTO CHAOS ***
20 REM *** JOYSTICK VERSION ***
40 GOSUB 1600:GOSUB 1800
50 POKE 53280,0:POKE 53281,0:DD=56320
60 PRINT "J":S$="J"      "
```

```

65 REM CLR SCRN UP CRSR 8 SPACES
70 Q=7:SC=1024:X=14
80 FOR RR=1 TO 200:PRINT " "
82 REM REV ON BLUE 38 SPACES
85 IF S$=" " THEN 1500:REM UP CRSR 2 SPACES
90 Z=RND(1)
100 Q=Q+(Z<.5)-(Z>=.5)
110 IF Q>24 THEN Q=24
120 IF Q<4 THEN Q=4
130 PRINT TAB(Q+3) S$
140 POKE SC+X,90:POKE SC+54272+X,1
150 X1=X:GOSUB 5000:IF FR<>16 THEN FOR TD=1 TO
300:NEXT
155 IF JV=4 THEN X=X-1
157 IF JV=8 THEN X=X+1
170 IF PEEK(SC+X+40)=160 THEN 1000
180 M=M+10:NEXT:S$=LEFT$(S$(LEN(S$)-1)):GOTO 80
1000 H=H+1
1010 POKE VO,15:POKE AD,17:POKE SR,164:POKE WF,
33:POKE HF,3:POKE LF,155
1020 FOR TD=1 TO 500:NEXT TD:POKE VO,0:POKE WF,0
1030 FOR TD=1 TO 1000:NEXT TD:IF H=4 THEN GOTO
1200
1050 IF PEEK(SC+X-2)=32 THEN X=X-2:GOTO 1070
1060 X=X+2
1070 GOTO 180
1200 PRINT " ":PRINT TAB(10) "THE TUNNELS OF CHAOS
OVER ****"
1205 REM ORANGE DOWN CRSR
1210 PRINT TAB(2) "YOU PENETRATED";M/1000;
"KILOMETRES INTO"
1215 REM DOWN CRSR LT BLUE
1220 PRINT TAB(9) "THE TUNNELS OF CHAOS"
1225 REM PURPLE DOWN CRSR
1230 POKE VO,15:POKE AD,15:POKE WF,17
1240 FOR H1=60 TO 20 STEP -2:FOR L1=250 TO 10
STEP -5
1250 POKE HF,H1:POKE LF,L1:NEXT:NEXT
1260 POKE VO,0:POKE WF,0
1270 FOR TD=1 TO 1000:NEXT
1280 POKE 53280,11:POKE 53281,6:PRINT " "

```



```

1290 PRINT TAB(10) "ANOTHER GAME (Y/N)"
1295 REM DOWN CRSR
1300 GET A$:IF A$="" THEN 1300
1310 IF A$="Y" THEN RUN
1320 IF A$="N" THEN END
1330 IF A$<>"Y" AND A$<>"N" THEN 1300
1500 PRINT "J"
1505 REM CLR SCRN
1510 POKE VO,15:POKE AD,17:POKE SR,32:POKE WF,
    17
1520 H1=INT(20*RND(1)+2):L1=INT(40*RND(1)+10)
1530 POKE HF,H1:POKE LF,L1:FOR TD=1 TO 50:NEXT
    TD
1540 SS=SS+1:IF SS>8 THEN SS=0:POKE VO,0:POKE
    WF,0:GOTO 1560
1550 GOTO 1520
1560 PRINT TAB(12) "WELL DONE!!"
1565 REM DOWN CRSR YELLOW
1570 PRINT TAB(4) "YOU MADE IT TO THE BOTTOM
    OF THE"
1575 REM DOWN CRSR
1580 REM DOWN CRSR ORANGE
1590 GOTO 1210
1600 PRINT"J"
1605 REM CLR SCRN
1610 VO=54296:AD=54278:SR=54277:WF=54276:
    HF=54273:LF=54272
1700 RETURN
1800 PRINT"J":POKE 53280,11:POKE 53281,14:REM
    CLR SCRN
1810 GOSUB 2000:PRINT "":PRINT TAB(10)
    "DESCENT INTO CHAOS"
1815 REM HOME DOWN CRSR
1820 GOSUB 2010:PRINT "":PRINT TAB(10)
    "DESCENT INTO CHAOS"
1830 GOSUB 2020:PRINT "":PRINT TAB(10)
    "DESCENT INTO CHAOS"
1840 POKE VO,15:POKE AD,17:POKE SR,132:POKE
    WF,33
1850 H1=100:L1=200
1860 POKE HF,H1:POKE LF,L1:FOR TD=1 TO 125:NEXT

```

```

1870 H1=H1-5:L1=L1-10:XX=XX+1:IF XX=17 THEN XX=0
      :POKE VO,0:POKE WF,0:RETURN
1880 GOTO 1860
2000 FOR TD=1 TO 500:NEXT:PRINT "█":RETURN
2010 FOR TD=1 TO 500:NEXT:PRINT "▣":RETURN
2020 FOR TD=1 TO 500:NEXT:PRINT "■":RETURN
5000 JV=PEEK(DD):FR=JV AND 16:JV=15-(JVAND15)
5010 RETURN

```

```

10 REM *** DESCENT INTO CHAOS ***
20 REM *** KEYBOARD VERSION ***
40 GOSUB 1600:GOSUB 1800
50 POKE 53280,0:POKE 53281,0
60 PRINT "J":S$="J"
65 REM CLR SCRN UP CRSR 8 SPACES
70 Q=7:SC=1024:X=14
80 FOR RR=1 TO 200:PRINT "██"
82 REM REV ON BLUE 38 SPACES
85 IF S$="J" THEN 1500:REM UP CRSR 2 SPACES
90 Z=RND(1)
100 Q=Q+(Z<.5)-(Z>=.5)
110 IF Q>24 THEN Q=24
120 IF Q<4 THEN Q=4
130 PRINT TAB(Q+3) S$
140 POKE SC+X,90:POKE SC+54272+X,1
150 X1=X:KI=PEEK(197):IF KI=22 THEN FOR TD=1 TO
      300:NEXT
155 X=X+(KI=47)-(KI=44)
170 IF PEEK(SC+X+40)=160 THEN 1000
180 M=M+10:NEXT:S$=LEFT$(S$, (LEN(S$)-1)):GOTO 80
1000 H=H+1
1010 POKE VO,15:POKE AD,17:POKE SR,164:POKE
      WF,33:POKE HF,3:POKE LF,155
1020 FOR TD=1 TO 500:NEXT TD:POKE VO,0:POKE WF,0
1030 FOR TD=1 TO 1000:NEXT TD:IF H=4 THEN GOTO
      1200
1050 IF PEEK(SC+X-2)=32 THEN X=X-2:GOTO 1070
1060 X=X+2
1070 GOTO 180

```

```

1200 PRINT "J":PRINT TAB(10) "***** GAME
    OVER ****"
1205 REM CLR SCRN ORANGE DOWN CRSR
1210 PRINT TAB(2) "YOU PENETRATED";M/1000;
    "KILOMETRES INTO"
1215 REM DOWN CRSR LT BLUE
1220 PRINT TAB(9) "THE TUNNELS OF CHAOS"
1225 REM PURPLE DOWN CRSR
1230 POKE VO,15:POKE AD,15:POKE WF,17
1240 FOR H1=60 TO 20 STEP -2:FOR L1=250 TO 10
    STEP -5
1250 POKE HF,H1:POKE LF,L1:NEXT:NEXT
1260 POKE VO,0:POKE WF,0
1270 FOR TD=1 TO 1000:NEXT
1280 POKE 53280,11:POKE 53281,6:PRINT "J"
1285 REM CLR SCRN
1290 PRINT TAB(10) "ANOTHER GAME (Y/N)"
1295 REM DOWN CRSR
1300 GET A$:IF A$="" THEN 1300
1310 IF A$="Y" THEN RUN
1320 IF A$="N" THEN END
1330 IF A$<>"Y" AND A$<>"N" THEN 1300
1500 PRINT "J"
1505 REM CLR SCRN
1510 POKE VO,15:POKE AD,17:POKE SR,32:POKE WF,17
1520 H1=INT(20*RND(1)+2):L1=INT(40*RND(1)+10)
1530 POKE HF,H1:POKE LF,L1:FOR TD=1 TO 50:NEXT
1540 SS=SS+1:IF SS>8 THEN SS=0:POKE VO,0:POKE
    WF,0:GOTO 1560
1550 GOTO 1520
1560 PRINT TAB(12) "WELL DONE!!"
1565 REM DOWN CRSR YELLOW
1570 PRINT TAB(4) "YOU MADE IT TO THE BOTTOM
    OF THE"
1575 REM DOWN CRSR
1580 REM DOWN CRSR ORANGE
1590 GOTO 1210
1600 PRINT"J"
1605 REM CLR SCRN
1610 VO=54296:AD=54278:SR=54277:WF=54276:
    HF=54273:LF=54272
1700 RETURN

```

```

1800 PRINT "I":POKE 53280,11:POKE 53281,14:REM
    CLR SCRN
1810 GOSUB 2000:PRINT "XXXXXXXXXX":PRINT TAB(10)
    "DESCENT INTO CHAOS"
1815 REM HOME DOWN CRSR
1820 GOSUB 2010:PRINT "XXXXXXXXXX":PRINT TAB(10)
    "DESCENT INTO CHAOS"
1830 GOSUB 2020:PRINT "XXXXXXXXXX":PRINT TAB(10)
    "DESCENT INTO CHAOS"
1840 POKE VO,15:POKE AD,17:POKE SR,132:POKE
    WF,33
1850 H1=100:L1=200
1860 POKE HF,H1:POKE LF,L1:FOR TD=1 TO 125:NEXT
1870 H1=H1-5:L1=L1-10:XX=XX+1:IF XX=17 THEN XX=0
    :POKE VO,0:POKE WF,0:RETURN
1880 GOTO 1860
2000 FOR TD=1 TO 500:NEXT:PRINT "R":RETURN:
    REM RED
2010 FOR TD=1 TO 500:NEXT:PRINT "B":RETURN:
    REM BLUE
2020 FOR TD=1 TO 500:NEXT:PRINT "█":RETURN:
    REM BLACK

```

FIRE

Fire uses the sprite graphics of the Commodore 64 to produce a great 'Invaders' type game.

The alien invader moves across the screen dropping bombs on you. Each time he reaches the edge of the screen he drops down to the next level before coming back the other way. If one of the bombs hits your fire base, or if you let the alien craft get down to your level, you will be blown up and, of course, lose the game.

Use the "J" key and the "K" key to move your fire base left and right respectively. The "F" key fires your missiles.

To destroy the alien space craft you must hit it dead centre. Hitting the alien ship on the hull will score you points but won't blow it up. Each time you destroy an alien ship it will be replaced by another one, which will move more quickly across the screen. Each time you blow up an alien ship and advance to the next level a new moon will appear in the sky. The number of moons indicates the level at which you are playing.

There is also a feature which gives you the option of playing with a background of mountains or in the blackness of space.

The program uses a very clever countdown routine to provide a touch of excitement to the start of the game.

The variables X and Y are used for the screen coordinates of your fire base and SX and SY are used to move the alien craft.

Fire was written by Daniel Quinn.

```

1 DIM S1(39):V=53248:HI=0:X$="NO ONE"
2 RESTORE:GOSUB 3000:POKE 53280,0:SC=0
3 DATA 1,2,2,2,2,1,3,3,3,3:REM THIS
4 DATA 1,2,2,2,2,1,2,2,2,1:REM DATA
5 DATA 1,3,3,3,3,3,3,3,3,3:REM IS
6 DATA 1,4,4,4,4,1,2,2,2,1:REM FOR
7 DATA 1,4,4,4,4,1,3,3,3,1:REM THE
8 DATA 2,2,2,2,2,1,3,3,3,3:REM COUNT-
9 DATA 1,3,3,3,3,1,3,3,3,1:REM DOWN
10 DATA 1,3,3,3,3,1,4,4,4,1
11 DATA 5,5,5,5,5,5,5,5,5,1,2,2,2,2,2,2,2,2,1
12 X=175:Y=195:P=195:SY=50:SP=SY:SM=SX:M=0:N=0:
    SE=1
23 FOR A=1 TO 70:K=INT(RND(1)*15):PRINT SPC(K)
    ;".":NEXT A
24 IF BG=0 THEN FOR A=0 TO 39:READ S1(A):NEXT:
    GOTO 41
25 FOR A=0 TO 39
26 READ Y1:S1(A)=Y1
27 FOR B=Y1 TO 19:POKE (55296+A+40*B),12
28 POKE (1024+A+40*B),93
30 NEXT
32 POKE (1024+A+40*Y1),100
40 NEXT
41 POKE 1164,81: POKE 55436,7
92 DATA 13,14,15,16,17,16,14,13,12,11,10,9,8,8,
    7,7,7,8,9,10,11,11,12,13,13,14
93 DATA 14,14,13,12,11,11,10,9,9,8,7,6,7,8
100 V=53248
130 POKE 2041,15:REM S-1=ALIEN BOMB
140 POKE 2042,13:REM S-2=ALIEN
150 POKE 2043,14:REM S-3=YOUR BASE
160 POKE 2044,15:REM S-4=YOUR MISSILE
170 POKE 2045,11:REM S-5=EXPLOSION
200 FOR N=0 TO 62:READ Q:POKE 832+N,Q:NEXT:REM
    POKEING DATA FOR ALIEN
210 DATA 0,0,0,0,90,0,0,255,0,1,129,128, 15,255,
    240,56,0,28,127,255,254
220 DATA 79,231,242,96,0,6,100,146,86,48,0,12,
    15,255,240,15,60,240
230 DATA 0,219,0,1,16,128,2,16,64,4,16,32,8,16,
    16,4,8,32,2,8,64,4,8,32

```

```

300 FOR N=0 TO 62:READ Q:POKE 896+N,Q:NEXT:REM
    POKEING DATA FOR YOUR BASE
320 DATA 0,0,0,0,0,0,0,129,0
330 DATA 1,129,128,0,129,0,0,129,0
340 DATA 1,129,128,3,0,192,3,129,192
350 DATA 15,0,240,31,0,248,43,0,212
360 DATA 33,255,132,33,255,132,34,255,68,31,126
    ,248,39,189,228,47,214,244
370 DATA 31,231,248,63,231,252,127,195,254
375 FOR N=704 TO 766:POKE N,255:NEXT:REM
    POKEING DATA FOR EXPLOSION
400 FOR A=0 TO 62:READ D:POKE 960+A,D:NEXT
410 DATA 0,24,0,0,24,0,0,36,0
420 DATA 0,66,0,0,126,0,0,126,0
430 DATA 0,126,0,0,66,0,0,66,0
440 DATA 0,60,0,0,60,0,0,36,0
450 DATA 0,36,0,0,60,0,0,60,0
460 DATA 0,0,0,0,0,0,0,74,0
470 DATA 0,82,0,0,36,0,0,24,0
473 POKE V+21,28:POKE V+41,1:POKE V+42,8:POKE
    V+43,14:POKE V+7,Y:N=0
498 POKE V+23,12:B=10:SI=0.5
500 I=PEEK(197)
501 PRINT"SCORE ";S:REM "罾" IS HOME
510 IF I=34 THEN X=X-B:GOTO530
520 IF I=37 AND X<244 THEN X=X+B:GOTO530
525 IF I=21 AND N=0 THEN P=P-1:L=X:N=2
530 IF P<195 AND P>10 THEN P=P-20:GOTO600
535 IF P<20 THEN L=X:N=0:P=195
560 IF P>190 THEN L=X
600 IF SX<X-10 OR SX>X+10 OR M=2 THEN 710
620 SP=SY:M=2:POKE V+21,30:SM=SX
710 IF M=2 THEN SP=SP+15
711 IF SP>220 OR M=0 THEN M=0:SP=SY:SM=SX:POKE
    V+21,28
740 IF X>255 THEN X=X-10
741 IF X<30 THEN X=X+10
750 POKE V+6,X
760 IF P<196 AND P>20 THEN POKE V+9,P
770 POKE V+8,L
771 O=SI*B:IF O=2 THEN O=-O
772 SX=SX+O

```

```

775 IF SX>254 THEN SY=SY+15:Q=2: SX=254
776 IF SX<30 THEN Q=0: SX=30: SY=SY+13
780 POKE V+4, SX: POKE V+5, SY
781 POKE V+2, SM: POKE V+3, SP
790 IF PEEK(V+30)=18 AND P<>195 THEN 800
791 IF PEEK(V+30)=12 OR PEEK(V+30)=28 OR PEEK
(V+30)=30 THEN 2000
795 IF PEEK(V+30)=20 OR PEEK(V+30)=22 THEN 900
796 IF PEEK(V+30)=10 OR PEEK(V+30)=18 AND P=195
OR PEEK(V+30)=26 THEN 2000
797 GOTO 850
800 POKE V+40, 2: POKE V+21, 62: POKE V+10, L: POKE
V+11, P
810 P=195: L=X: SP=SY: M=0: N=0: POKE V+44, 1
820 FOR A=1 TO 100: NEXT: POKE V+21, 30
830 POKE V+44, 1
850 GOTO 500
900 IF L>SX-2 AND L<SX+2 THEN 909
901 POKE V+10, L: POKE V+11, SY: POKE V+21, 62
902 FOR A=1 TO 50
903 POKE V+44, 1: POKE V+44, 10
904 NEXT A: POKE V+21, 14
907 SC=SC+10
908 GOTO 500
909 IF L=SX THEN SC=SC+100
910 POKE V+21, 0
911 PRINT "J": FOR A=1 TO 30
912 REM CLR SCRN
920 IF A/2=INT(A/2) THEN C=2: B=0: GOTO 940
930 C=0: B=2
940 POKE 53280, C
950 POKE 53281, B
960 NEXT
965 SC=SC+100
980 POKE 53280, 0: POKE 53281, 0
1000 X=175: Y=195: P=195: SY=50: SX=20: SP=SY: SM=SX:
M=0: N=0: Q=0: SE=SE+1
1001 FOR A=1 TO 70: K=INT(RND(1)*15): PRINT SPC
(K): ". ": NEXT A
1002 FOR TY=1 TO SE: WE=INT(388*RND(1)+1024):
POKE WE, 81: POKE WE+54272, 7: NEXT TY
1003 SI=SI+0.5

```



```

1004 IF BG=0 THEN 1010
1005 FOR A=0 TO 39:FOR B=S1(A) TO 19
1007 POKE (1024+A+40*B),224
1008 POKE (55296+A+40*B),3:NEXT
1009 POKE (1024+A+40*S1(A)),88:NEXT
1010 B=10:GOTO 500
2000 POKE V+21,0:PRINT"GAME OVER" :REM CLR YELLOW
      DOWN CRSR
2020 PRINT "#####"
2030 PRINT "#####"
2040 PRINT "#####"
2050 PRINT "#####"
2060 PRINT "#####"
2070 PRINT "#####"
2080 PRINT "#####"
2085 REM THE ABOVE CHARACTER IS CHR$(166)
2090 PRINT:PRINT:PRINT SPC(15);"GAME OVER"
      :REM RVS ON BLUE WHITE REVS OFF
2091 PRINT:PRINT TAB(10);"HI SCORE:";HI:PRINT
      TAB(10);"HELD BY:";X$
2100 PRINT"DOWN CRSR"
2110 PRINT TAB(10);"YOUR SCORE ";SC:REM YELLOW
2111 IF SC<=HI THEN 2120
2112 PRINT TAB(10);"CONGRATULATIONS!!!": REM
      PURPLE
2113 PRINT TAB(10);"YOU'RE THE NEW CHAMP":PRINT
      TAB(10);"WHAT IS YOUR NAME";
2114 REM YELLOW
2115 INPUT X$:HI=SC
2120 FOR Z=1 TO 500:NEXT
2130 PRINT"WHITE" :REM CLR WHITE
2140 PRINT" DO YOU WISH TO:" :PRINT
2150 PRINT" (1)..END GAME"
2160 PRINT" (2)..PLAY AGAIN"
2170 GET K$:IF VAL(K$)=1 THEN END
2180 IF VAL(K$)=2 THEN GOTO 2
2190 GOTO 2170
3000 PRINT " "
3010 REM CLR SCRN
3100 POKE 53280,0:POKE 53281,0
3155 PRINT " "
3160 REM CLR SCRN

```

```

3500 N$(1)= "  "
3505 REM REV ON
3510 N$(2)= "  ██████████ "
3515 REM RIGHT CRSR
3520 N$(3)= " ██████████ "
3530 N$(4)= "  ██████████ "
3540 N$(5)= " ██████████ ██████████ "
3545 FOR A=9 TO 0 STEP -1
3560 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:
      PRINT"  "
3565 REM WHITE
3570 FOR Z=1 TO 10
3580 READ F:PRINT TAB(15);"  ";N$(F)
3585 REM ORANGE
3590 NEXT
3600 PRINT "  ":REM CLR SCRN
3601 FOR Q=1 TO 100:NEXT
3610 NEXT
3615 FOR Q=1 TO 100:NEXT
3620 PRINT "  "
3625 REM CLR SCRN
3630 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT"  "
3635 REM YELLOW
3640 PRINT "      ████████ ███ ████████ ████████ ███ ███"
3650 PRINT "      ██████████ █████ ███ ██████████ ██████████ ██████████"
3660 PRINT "      ██████████ █████ ██████████ ██████████ ██████████ ██████████"
3670 PRINT "      ██████████ █████ ██████████ ██████████ ██████████ ██████████"
3680 PRINT "      ██████████ █████ ██████████ ██████████ ██████████ ██████████"
3690 PRINT "      ██████████ █████ ██████████ ██████████ ██████████ ██████████"
3700 PRINT "      ██████████ █████ ██████████ ██████████ ██████████ ██████████"
3701 PRINT:PRINT:PRINT TAB(10);"J..LEFT      K.
      .RIGHT"
3702 PRINT:PRINT TAB(10);"      F..FIRE      ":
      PRINT
3710 FOR A=1 TO 30:IF A/2=INT(A/2) THEN C=2:B=0:
      GOTO 3720
3715 C=0:B=10
3720 POKE 53290,C:POKE 53281,B
3740 NEXT
3745 PRINT "      DO YOU WANT A BACKGROUND?"
3746 GET N$:IF N$="Y" THEN BG=1:GOTO 3749
3747 IF N$="N" THEN BG=0:GOTO 3749
3748 GOTO 3746

```

```
3749 PRINT "J": REM CLR SCRN
3750 RETURN
```

SKIING

This program by William Leader gives you the opportunity to try out your skills as a downhill skier.

The program will give you a choice of a slalom course or a jumps course to provide variety. The object of the game is to complete the course with as few collisions, and taking as many jumps or flag gates, as possible. Your final score is based on your time plus penalties for missed jumps or gates and collisions. So, of course, the lower the score the better you've done.

On both the slalom and the jumps course, your skier will appear on screen at the start of the game standing underneath the starting banner. Use either the joystick or the relevant keys to move him into the starting location you require and then press the fire button (or the space bar) to start the run.

On the slalom course, you will be required to avoid hitting the trees dotted all over the place and to steer your man through the flag gates. You can use the controls to move your skier to the left and right and up and down the slope.

On the jumps course the controls work just the same as on the slalom course, but to successfully take your man over one of the jumps the fire button (or the spacebar) must be pressed just before you reach the jump. On both courses the screen will briefly flash dark blue to indicate that the obstacle has been negotiated safely.

There are two listings of the game. The first is the joystick-operated version and the second is for a keyboard. In the keyboard version the controls are:-

"K" = up

"M" = down

"Z" = left

"X" = right

Spacebar = jump and start game.

Lines 1 to 11 are an initialisation routine. They allow you to select the type of course and the number of people who are going to play the game. Line 12 provides the prompt message for the start

of play. Lines 13 to 16 PRINT up the start line using graphic characters from the keyboard. Lines 21 to 24 are further initialisation routines.

The two FOR statements in lines 25 and 26 are the start of the two FOR/NEXT loops which control the frequency of the flags and the jumps during the running of the program. Line 27 reads the joystick or keyboard input and line 28 controls the jumping routine. If a jump is in progress the routine which changes the screen location values in the next few lines is bypassed.

Line 29 uses the value of TI\$, the internal clock, to PRINT the score on the screen. Lines 30 to 38 use the input to change the values which are used to POKE your skier onto the screen.

Line 29 is also part of the routine which controls the jumping. Lines 40 to 60 are safeguards to prevent your skier wandering off the screen during the course of the game.

The sprites used in the game are POKEd onto the screen by lines 100 to 108. The lines 114 to 116 check to see if the fire button or spacebar has been pressed to start the game.

Line 117 reads the value of the fire button or the spacebar and creates the jump if it has been pressed. Lines 118 and 119 check to see if the skier has passed the jump. Line 120 checks for a collision.

Lines 122 to 168 move the trees smoothly up the screen and differentiate between the slalom and the jumps course. Line 170 sends us around the loop again.

Lines 180 to 186 call up the new flags or jump and line 190 calls up the end routine when the course has been run. Lines 195 to 340 contain the DATA statements for the sprites. The sprites are initialised in lines 4000 to 4060. The crash routine is in lines 5000 to 5100. This is called up by line 120.

The finish line routine uses graphics characters from the keyboard to PRINT up the display. The crowd is made up from the symbols on the "W", "Q" and "E" keys.

```

1 REM *** SKIING JOYSTICK VERSION ***
2 PRINT"0":BH=20:VH=10:RU=3:CR=6:TI$="000100":
  REM CLR SCRN
3 PRINT"SLALOM OR JUMP(S/J)":INPUT A$
4 PRINT"HOW MANY PLAYERS?":INPUT S
5 DIM AK(S):PRINT"PLEASE WAIT"
6 POKE 53281,1:POKE 53280,14:R=0
7 V=50246:POKE V+21,127
8 POKE 2040,192:POKE 2041,193:POKE 2042,194:
  POKE 2043,195:POKE 2044,196
9 POKE 2045,197:POKE 2046,198:POKE 2047,199
10 GOTO 4000
11 FOR S1=12416 TO 12478:READ Q:POKE S1,Q:NEXT
  :GOTO 4055
12 FOR IA=1 TO S:PRINT"PLAYER"IA"HIT RETURN TO
  PLAY#":INPUT IOU$
13 PRINT"00":PRINT TAB(12)"|-----|":
  REM CLR SCRN LT BLUE
14 PRINT TAB(12)"|THE SKI CHMP$ 0 1":REM LT
  RED LT BLUE
15 PRINT TAB(12)"|-----|
16 PRINT TAB(12)"|
20 GOTO 21
21 X=220:Y=80:R=20:I=100:N=150:O=200:M=250:ER=
  75:RU=RU+1:POKE V+21,127
22 FOR TYU=39 TO 45:POKE V+TYU,5:NEXT:POKE V+41
  ,RU:IF A$="S" THEN BH=10:VH=20
23 POKE 53281,1:G=90:DF=20*RND(1)+55
24 L=250*RND(1):K=150*RND(1)+100:H=205*RND(1):
  T=50*RND(1)+200
25 FOR RF=1 TO VH
26 FOR CF=1 TO BH
27 JV=PEEK(56320):FR=JVAND16:JV=15-(JVAND15)
28 IF JUM=100 THEN VB=VB+1:GOTO 39
29 PRINT"00":INT(VAL(TI$)):REM HOME
30 IF JV=4 THEN XYZ=-8
31 IF JV=2 THEN Y=Y+8
32 IF JV=1 THEN Y=Y-8
33 IF JV=8 THEN XYZ=8
38 X=X+XYZ
39 IF VB=6 THEN JUM=0:VB=0:POKE V+41,RU

```

```

40 IF X<5 THEN X=5:XYZ=8
50 IF X>250 THEN X=250:XYZ=-8
55 IF Y>240 THEN Y=240
60 IF Y<50 THEN Y=50
100 POKE V+4,X:POKE V+5,Y
101 POKE V+6,L:POKE V+7,R
102 POKE V+8,K:POKE V+9,I
103 POKE V+10,H:POKE V+11,N
104 POKE V+12,T:POKE V+13,O
105 POKE V+2,G:POKE V+3,M
106 POKE V,DF:POKE V+1,ER
107 IF QY<10 THEN 114
108 POKE V+14,QX:POKE V+15,QY
114 IF P=2 THEN 117
115 IF FRC>16 THEN P=2:TI$="000100":PRINT"J":
    FR=16:REM CLR SCRN
116 IF P<>2 THEN 25
117 IF FRC>16 THEN JUM=100:POKE V+41,3
118 IF A$="S" AND (QX>X-10 AND QX<X+10) AND
    (QY>Y-15 AND QY<Y+15) THEN 5500
119 IF (QX>X-10 AND QX<X+10) AND (QY>Y-15 AND
    QY<Y+15) AND JUM=100 THEN 5500
120 IF PEEK(53248+30)AND4=4 THEN 5000
122 WES=20:R=R-WES:I=I-WES:N=N-WES:O=O-WES:
    M=M-WES:ER=ER-WES:QY=QY-WES
129 IF A$="S" THEN 157
130 IF RC1 THEN R=240:L=20*RNDRND(1)+50
135 IF IC1 THEN I=240:K=30*RNDRND(1)
140 IF NC1 THEN N=240:H=20*RNDRND(1)+80
145 IF OC1 THEN O=240:T=100*RNDRND(1)+135
150 IF MC1 THEN M=240:G=100*RNDRND(1)+150
155 IF EC1 THEN ER=240:DF=100*RNDRND(1)+150
156 GOTO 169
157 IF RC1 THEN R=240:L=20
158 IF IC1 THEN I=240:K=35
159 IF NC1 THEN N=240:H=55
160 IF OC1 THEN O=240:T=180
161 IF MC1 THEN M=240:G=210
162 IF EC1 THEN ER=240:DF=245
169 IF QY<1 THEN POKE V+21,127
170 NEXT
180 POKE V+21,255:QX=120:QY=240

```

```

185 IF A$="S" THEN QX=85*RND(1)+75
186 NEXT
190 GOTO 6000
195 DATA 0,24,0,0,24,0,0,60,0,0,126,0,0,255,0,
      1,255,128,1,255,128
196 DATA 3,255,192,3,255,192,7,255,224,6,255,96,
      0,255,0,1,255,128
197 DATA 3,255,192,3,255,192,7,255,224,15,189,
      240,12,60,48
198 DATA 0,60,0,0,60,0,0,60,0
200 DATA 0,48,0,0,72,0,0,72,0,0,48,0,0,120,0,1,
      134,0,3,142,0,6,156,32
210 DATA 12,191,240,24,104,32,16,249,0,17,153,0,
      19,50,0,19,116,0
220 DATA 59,184,0,19,48,0,2,32,0,4,64,0,8,128,
      0,85,0,0,34,0,0
300 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
      0,0,0,0,0,4,0,32,10,0,64
310 DATA 19,255,192,1,255,144,28,145,96,2,110,
      128,1,0,0,0,48,0,0,0,0
320 DATA 0,0,0,0,0,0,0,0,0,0,0,0
330 DATA 0,0,0,0,0,0,128,0,1,128,0,1,128,0,1,
      128,0,1,128,0,1,128,0,1
335 DATA 128,0,1,128,0,1,128,0,1,128,0,1,128,0,
      1,128,0,1,192,0,6,0,0,0
340 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
      0,0,0,0,0,0,0,0,0,0,0,0
4000 FOR S2=12388 TO 12350:READ Q2:POKE S2,Q2:
      NEXT:RESTORE
4010 FOR S3=12352 TO 12414:READ Q3:POKE S3,Q3:
      NEXT:RESTORE
4020 FOR S4=12480 TO 12542:READ Q4:POKE S4,Q4:
      NEXT:RESTORE
4030 FOR S5=12544 TO 12606:READ Q5:POKE S5,Q5:
      NEXT:RESTORE
4040 FOR D2=12608 TO 12670:READ W2:POKE D2,W2:
      NEXT:RESTORE
4050 FOR D3=12672 TO 12734:READ W3:POKE D3,W3:
      NEXT:GOTO 11
4055 FOR D4=12736 TO 12798:READ W4:POKE D4,W4:
      NEXT

```



```

4056 IF A$="S" THEN FOR D4=12736 TO 12798:READ
      W4:POKE D4,W4:NEXT
4060 GOTO 12
5000 IF CR=6 THEN CR=1:GOTO 122
5010 S=54272:FOR L=0 TO 24:POKE S+L,0:NEXT:POKE S
      +1,100:POKE S+5,10 :POKE S+24,15
5020 POKE S+4,129:SC=SC-40
5100 POKE V+41,0:FOR P0=1 TO 500:NEXT:Y=Y+15:
      POKE V+5,Y:POKE V+41,RU:CR=6:GOTO 122
5500 SC=SC+20
5510 POKE 53281,6:FOR TR=1 TO 10:Y=Y+1:POKE V+5,
      Y:NEXT:POKE 53281,1:CR=6:GOTO 122
6000 FOR W=1 TO 20:PRINT:NEXT
6005 REM "||" IS RIGHT CRSR
6010 PRINT"|||||r-----"
6020 PRINT"|||||THE FINISH LINE...||"
6025 REM YELLOW LT BLUE
6030 PRINT"|||||-----"
6040 PRINT"|||||"
6041 PRINT"TTTTTTT":PRINT TAB(25)" 0 0 000":
      REM UP CRSR
6042 PRINT TAB(25)" +000 000 0000"
6043 PRINT TAB(25)" ||| ||| ||||"
6044 PRINT TAB(25)"0000 00 000000"
6045 PRINT TAB(25)"|||| || ||||"
6050 REM CLR SCRN
6051 IF JV=4 THEN X=X-5
6052 IF JV=8 THEN X=X+5
6053 Y=Y+4
6054 IF Y>240 THEN GOTO 6058
6055 IF X>250 THEN X=250
6056 IF X<25 THEN X=25
6057 POKE V+4,X:POKE V+5,Y:GOTO 6050
6058 RE=INT(VAL(TI$))-SC
6060 FOR Y=1 TO 3000:NEXT:POKE V+21,0:PRINT"||":
      PRINT"SCORE IS:")RE
6070 AK(IA)=RE:FOR E=1 TO IA:PRINT"PLAYER":E:
      "SCORE=":AK(E):NEXT
6080 TI$="000000":P=0:CR=6:XYZ=0:SC=0:RE=0
6090 NEXT IA
7000 FOR T=1 TO 1000:NEXT:PRINT"|||||DOES
      ANYONE WANT TO PLAY AGAIN?"

```



```

26 FOR GF=1 TO BH
27 JV=PEEK(197):FR=16:IF JV=60 THEN FR=10
28 IF JUM=100 THEN VB=VB+1:GOTO 39
29 PRINT"8":INT(VAL(TI$)):REM HOME
30 IF JV=12 THEN XYZ=-9
31 IF JV=36 THEN Y=Y+9
32 IF JV=37 THEN Y=Y-9
33 IF JV=23 THEN XYZ=9
38 X=X+XYZ
39 IF VB=6 THEN JUM=0:VB=0:POKE V+41,RU
40 IF X<5 THEN X=5:XYZ=8
50 IF X>250 THEN X=250:XYZ=-8
55 IF Y>240 THEN Y=240
60 IF Y<50 THEN Y=50
100 POKE V+4,X:POKE V+5,Y
101 POKE V+6,L:POKE V+7,P
102 POKE V+8,K:POKE V+9,I
103 POKE V+10,H:POKE V+11,N
104 POKE V+12,T:POKE V+13,O
105 POKE V+2,G:POKE V+3,M
106 POKE V,DF:POKE V+1,ER
107 IF QY<10 THEN 114
108 POKE V+14,QX:POKE V+15,QY
114 IF P=2 THEN 117
115 IF FRC>16 THEN P=2:TI$="000100":PRINT"J":
    FR=16:REM CLR SCRN
116 IF P<>2 THEN 25
117 IF FRC>16 THEN JUM=100:POKE V+41,3
118 IF A$="S" AND (QX>X-10 AND QX<X+10) AND
    (QY>Y-15 AND QY<Y+15) THEN 5500
119 IF (QX>X-10 AND QX<X+10) AND (QY>Y-15 AND
    QY<Y+15) AND JUM=100 THEN 5500
120 IF PEEK(53248+30)AND4=4 THEN 5000
122 WES=20:R=R-WES:I=I-WES:N=N-WES:O=O-WES:M=M-
    WES:ER=ER-WES:QY=QY-WES
129 IF A$="S" THEN 157
130 IF R<1 THEN R=240:L=20*RND(1)+50
135 IF I<1 THEN I=240:K=30*RND(1)
140 IF N<1 THEN N=240:H=20*RND(1)+80
145 IF O<1 THEN O=240:T=100*RND(1)+135
150 IF M<1 THEN M=240:G=100*RND(1)+150
155 IF ER<1 THEN ER=240:DF=100*RND(1)+150

```

```

156 GOTO 169
157 IF RC1 THEN R=240:L=20
158 IF IC1 THEN I=240:K=35
159 IF NC1 THEN N=240:H=55
160 IF OC1 THEN O=240:T=180
161 IF MC1 THEN M=240:G=210
162 IF ER1 THEN ER=240:DF=245
169 IF QYC1 THEN POKE V+21,127
170 NEXT
180 POKE V+21,255:QX=120:QY=240
185 IF A$="S" THEN QX=85*RND(1)+75
186 NEXT
190 GOTO 6000
195 DATA 0,24,0,0,24,0,0,60,0,0,126,0,0,255,0,1,
      255,128,1,255,128
196 DATA 3,255,192,3,255,192,7,255,224,6,255,96,
      0,255,0,1,255,128
197 DATA 3,255,192,3,255,192,7,255,224,15,189,
      240,12,60,48
198 DATA 0,60,0,0,60,0,0,60,0
200 DATA 0,48,0,0,72,0,0,72,0,0,48,0,0,120,0,1,
      134,0,3,142,0,6,156,32
210 DATA 12,191,240,24,104,32,16,249,0,17,153,
      0,19,50,0,19,116,0
220 DATA 59,194,0,19,48,0,2,32,0,4,64,0,8,128,
      0,85,0,0,34,0,0
300 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
      0,0,0,0,0,4,0,32,10,0,64
310 DATA 19,255,192,1,255,144,28,145,96,2,110,
      128,1,0,0,0,48,0,0,0,0
320 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
330 DATA 0,0,0,0,0,0,128,0,1,128,0,1,128,0,1,
      128,0,1,128,0,1,128,0,1
335 DATA 128,0,1,128,0,1,128,0,1,128,0,1,128,0,
      1,128,0,1,192,0,6,0,0,0
340 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
4000 FOR S2=12288 TO 12350:READ Q2:POKE S2,Q2:
      NEXT:RESTORE
4010 FOR S3=12352 TO 12414:READ Q3:POKE S3,Q3:
      NEXT:RESTORE

```

```

4020 FOR S4=12480 TO 12542:READ Q4:POKE S4,Q4:
    NEXT:RESTORE
4030 FOR S5=12544 TO 12606:READ Q5:POKE S5,Q5:
    NEXT:RESTORE
4040 FOR D2=12608 TO 12670:READ W2:POKE D2,W2:
    NEXT:RESTORE
4050 FOR D3=12672 TO 12734:READ W3:POKE D3,W3:
    NEXT:GOTO 11
4055 FOR D4=12736 TO 12798:READ W4:POKE D4,W4:
    NEXT
4056 IF A$="S" THEN FOR D4=12736 TO 12798:READ
    W4:POKE D4,W4:NEXT
4060 GOTO 12
5000 IF CR=6 THEN CR=1:GOTO 122
5010 S=54272:FOR L=0 TO 24:POKE S+L,0:NEXT:POKE S
    +1,100:POKE S+5,10 :POKE S+24,15
5020 POKE S+4,129:SC=SC-40
5100 POKEV+41,0:FOR P0=1 TO 500:NEXT:Y=Y+15:
    POKEV+5,Y:POKEV+41,RU:CR=6:GOTO122
5500 SC=SC+20
5510 POKE53281,6:FOR TR=1 TO 10:Y=Y+1:POKEV+5,Y
    :NEXT:POKE53281,1:CR=6:GOTO 122
6000 FOR W=1 TO 20:PRINT:NEXT
6005 REM "M" IS RIGHT CRSR
6010 PRINT"#####"
6020 PRINT"#####THE FINISH LINE...!"
6025 REM YELLOW LT BLUE
6030 PRINT"#####|
6040 PRINT"#####|
6041 PRINT"TTTTTTT":PRINT TAB(25)" 0 0 000":
    REM UP CRSR
6042 PRINT TAB(25)" -000 000 0000"
6043 PRINT TAB(25)"  III III  IIII"
6044 PRINT TAB(25)"#### 00 00####"
6045 PRINT TAB(25)"IIII II  IIII"
6050 REM CLR SCRN
6051 IF JV=4 THEN X=X-5
6052 IF JV=8 THEN X=X+5
6053 Y=Y+4
6054 IF Y>240 THEN 6058

```

```

6055 IF X>250 THEN X=250
6056 IF X<25 THEN X=25
6057 POKE V+4,X:POKE V+5,Y:GOTO 6050
6058 RE=INT(VAL(TI$))-SC
6060 FOR Y=1 TO 3000:NEXT:POKE V+21,0:PRINT "J":
      PRINT "SCORE IS: ";RE
6070 AK(IA)=RE:FOR E=1 TO IA:PRINT "PLAYER";E;
      "SCORE=";AK(E):NEXT
6080 TI$="000000":P=0:CR=6:XYZ=0:SC=0:RE=0
6090 NEXT IA
7000 FOR T=1 TO 1000:NEXT:PRINT "DOES
      ANYONE WANT TO PLAY AGAIN?"
7005 REM DOWN CURSR GREEN
7010 GET A$:IF A$<>"Y" THEN 7020
7015 RUN
7020 IF A$<>"N" THEN 7010
7030 END

```

SPOTLIGHT

Here we have another great clever program written by William Leader. You are a guard on the prison planet Uriel. This harsh, bleak planet is being used to hold a vicious band of space pirates. You are on duty in your guard tower the night the pirates plan to break out of the prison, climb the prison wall next to the space port and steal a ship to leave the planet. Your duty is to prevent the prisoners from reaching the wall. You must use your spotlight to patrol the pitch black yard and, if you see an escaping prisoner, fire your blaster to prevent him reaching the wall.

The game can be played with either a joystick or by using the keyboard. When you are using a joystick, the fire button fires your blaster. In the keyboard version of the game, the "K" key moves the spotlight up the screen and the "M" key moves it down screen. The "Z" and the "X" keys move it left and right respectively. The space bar fires the blaster.

There are five levels of difficulty in the game. Level one is the easiest and level five will provide a real challenge for the best of the arcade champions! The level of difficulty relates to how fast the escaping prisoners move. After you have stopped twenty prisoners from escaping, you will automatically progress to the next level.

When the program is RUN you will be asked to indicate which level of difficulty you wish to play. After you have answered this question you will be asked if you will use a joystick or the keyboard. The screen will then go black, the wall will be printed across the top of the screen and the beam from your spotlight will appear on the screen.

There will then be a short delay while the program organises itself. A siren will then sound, warning you of an escape attempt and, in the top left hand corner of the screen, you will be shown how many prisoners are escaping. It is a good idea to keep a count of the number of prisoners you have stopped as this prevents you from accidentally missing one and allowing him to escape.

You must start searching for the escaping pirates. When you find one, press the firing button on your blaster to stop him. You will see that there are other objects in the prison yard as well as the prisoners. Don't let them distract you. You will have no trouble recognising the pirates, they are dark shadowy figures who carry long, curved knives. They are very dangerous!

Lines 1 to 24 initialise the variables and set up the parameters for the program. They also ask for the level of difficulty and whether or not you will use a joystick. Line 21 prints the wall.

Line 25 reads the joystick input. Line 26 goes to the routine which advances levels. Here a check is made to see if a new level has been reached and provides the new variables and tells you the next level has been reached, if required.

Line 27 calls up the routine which reads the keyboard input if the keyboard is being used. Lines 30 to 37 act upon the joystick values, by altering the variables which are used to control the screen position of the spotlight. If the fire button has been pressed, line 38 calls up the sound routine.

Lines 40 and 41 enable the sprite which is used for the spotlight to cross the 255th screen location and move all the way to the right hand edge of the screen. Lines 42 to 45 make sure that your spotlight beam stays on the screen.

Line 46 checks to see if there is an escape in progress and if there isn't, goes and starts one. Lines 50 to 53 move the escapees randomly up the screen. Line 60 checks to see if prisoner number one has escaped and lines 61 to 63 make sure that the prisoner stays on the screen. Lines 65 to 68 perform the same function for prisoner number two.

Lines 100 and 110 POKE the sprites onto the screen and lines 118 and 120 check to see if a prisoner has been hit. Line 130 sends the program around again.

Lines 200 to 280 contain the DATA for the sprites. We check to see if a prisoner has been killed in lines 400 to 420. If he has, he is removed and is replaced by lines 430 and 435.

Lines 450 to 460 check to see if the prisoners have staged a breakout. Lines 465 and 475 warn you of the escape.

The buildings are printed on the screen by lines 500 to 590 and the sound routines are contained in lines 600 to 650.

Lines 800 to 870 tell you that a prisoner has made it over the wall, they also advise you of your score and ask if you wish to play again.

Lines 900 to 920 advance the program to the next level of difficulty and the keyboard reading routine is in lines 1000 to 1015.

```
1 POKE 53281,5:POKE 53280,5:PRINT"#####  
  WHAT LEVEL(1-5)?"  
2 GET A$:IF A$="" OR VAL(A$)<1 OR VAL(A$)>5  
  THEN 2  
3 L=VAL(A$):LV=L*4:PL=L:L=L*2:ES=0:SC=0:PRINT  
  "J":REM CLR SCRN  
5 POKE 53275,255:AW=SC  
7 X2=200*RND(1)+50:Y2=230:X=100:Y=100  
8 X3=200*RND(1)+50:Y3=230  
9 POKE 53281,0:POKE 53280,0  
10 V=53248  
12 POKE 2040,192:POKE 2041,193:POKE 2047,199  
13 PRINT"#####KEYBOARD OR JOYSTICK(K/J)?":  
  REM WHITE DOWN CRSR  
14 GET K$:IF K$="" THEN 14  
15 PRINT"J":FOR I=1 TO 40:T=1000*RND(1):POKE 55296  
  +T,0:POKE 1024+T,88:NEXT:GOSUB 500  
16 FOR I=1 TO 10:T=1000*RND(1):POKE 55296+T,0:  
  POKE 1024+T,65:NEXT:POKE V+21,131  
17 FOR I=1 TO 40:T=1000*RND(1):POKE 55296+T,0:  
  POKE 1024+T,88:NEXT:GOSUB 500  
18 FOR N=12288 TO 12350:READ Q:POKE N,Q:NEXT  
19 RESTORE:FOR N=12352 TO 12414:READ Q:POKE  
  N,Q:NEXT  
20 FOR N=12736 TO 12798:READ Q:POKE N,Q:NEXT  
21 PRINT"#####  
  ###"  
22 REM HOME LT RED SHIFT + 3 KEY  
24 POKE V+39,0:POKE V+40,0:POKE V+46,1
```

```

25 JV=PEEK(56320):FR=JVAND16:JV=15-(JVAND15)
26 GOSUB 900
27 IF K$="K" THEN 1000
30 IF JV=1 THEN Y=Y-8
31 IF JV=2 THEN Y=Y+8
32 IF JV=4 THEN X=X-8
33 IF JV=8 THEN X=X+8
34 IF JV=5 THEN Y=Y-8:X=X-8
35 IF JV=6 THEN Y=Y+8:X=X-8
36 IF JV=9 THEN Y=Y-8:X=X+8
37 IF JV=10 THEN Y=Y+8:X=X+8
38 IF FR<>16 THEN GOSUB 620
40 IF X<255 AND PEEK(V+16)=0 THEN POKE V+16,
    128:X=1
41 IF X<1 AND PEEK(V+16)=128 THEN X=255:POKE
    V+16,0
42 IF Y<50 THEN Y=50
43 IF Y>229 THEN Y=229
44 IF X<0 THEN X=0
45 IF X<28 AND PEEK(V+16)=0 THEN X=28
46 IF ES=0 THEN POKE V+14,X:POKE V+15,Y:GOTO 460
50 X2=X2+INT(LV*RND(1)-(LV/2))
51 Y2=Y2-INT(L*RND(1))
52 X3=X3+INT(LV*RND(1)-(LV/2))
53 Y3=Y3-INT(L*RND(1))
60 IF Y2 <50 THEN POKE V+39,7:PRINT"J":GOTO 800
    :REM CLR SCRN
61 IF Y2>227 THEN Y2=227
62 IF X2<24 THEN X2=24
63 IF X2>250 THEN X2=250
65 IF Y3<50 THEN POKE V+40,7:PRINT"J":GOTO 800
    :REM CLR SCRN
66 IF Y3>227 THEN Y3=227
67 IF X3<24 THEN X3=24
68 IF X3>250 THEN X3=250
100 POKE V,X2:POKE V+2,X3:POKE V+14,X
110 POKE V+1,Y2:POKE V+3,Y3:POKE V+15,Y
118 IF FR=16 THEN 130
120 IF PEEK(V+30)AND4=4 THEN 400
130 GOTO 25
200 DATA 0,0,16,0,24,8,0,36,4,0,36,2,0,24,3,0,
    24,7,0,254,14,1,255,252

```

```

210 DATA 3,255,248,15,235,240,14,126,0,28,126,0
    ,56,60,0,112,62,0
220 DATA 224,119,0,64,231,0,1,199,0,2,139,0,7,7
    ,0,14,7,0,14,7,0
250 DATA 0,255,0,3,255,192,7,255,224,15,255,240
    ,31,255,248,63,255,252
260 DATA 127,255,254,255,255,255,255,255,255,
    255,255,255,255,255,255
270 DATA 255,255,255,255,255,255,255,255,255,
    127,255,254,63,255,252
280 DATA 31,255,248,15,255,240,7,255,224,3,255,
    192,0,255,0
400 IFX<X2+15ANDX>X2-15ANDY<Y2+15ANDY>Y2-15THEN
    X2=200*RND(1)+50:Y2=230:GOTO430
410 IFX<X3+15ANDX>X3-15ANDY<Y3+15ANDY>Y3-15THEN
    X3=200*RND(1)+50:Y3=230:GOTO430
420 GOTO 130
430 GOSUB 640:SC=SC+50:NU=NU-1:IF NU<1 THEN 450
435 GOTO 130
450 NU=0:ES=0:X2=0:Y2=0:X3=0:Y3=0
455 POKE V,0:POKE V+1,0:POKE V+2,0:POKE V+3,0:
    GOTO 130
460 CH=0:CH=INT(35*RND(1))+1:IF CH<>1 THEN 25
465 NU=INT(10*RND(1))+2:X2=200*RND(1)+1:X3=200*
    RND(1)+1:Y2=230:Y3=Y2
475 GOSUB600:PRINT"NO.=":NU:FORT=1TO600:NEXT
    :PRINT" "
    :ES=100:GOTO25
476 REM PUT CRSR BACK INTO LINE BEFORE PRESSING
    RETURN
500 PRINT" ":REM HOME BLACK DOWN CRSR
501 REM " " IS RIGHT CRSR

```

```

502 PRINT"#####"
505 PRINT"#####"
510 PRINT"#####"
515 PRINT"#####"
520 PRINT"#####"
525 PRINT"#####"
530 PRINT"#####"
535 PRINT"#####"
540 PRINT"#####"
545 PRINT"#####"
550 PRINT"#####"
555 PRINT"#####"
560 PRINT"#####"
565 PRINT"#####"
570 PRINT"#####"
590 RETURN
600 FOR F=1 TO 10
605 S=54272:FOR W=0 TO 24:POKE S+W,0:NEXT:POKE S+1,
    100:POKE S+5,15:POKE S+15,255
610 POKE S+24,15:POKE S+4,17:FOR T=1 TO 100:
    NEXT:POKE S+4,18:NEXT
615 RETURN
620 S=54272:FOR W=0 TO 24:POKE S+W,0:NEXT:POKE
    S+1,20:POKE S+5,5:POKE S+15,255
625 POKE S+24,15:POKE S+4,129
630 RETURN
640 S=54272:FOR W=0 TO 24:POKE S+W,0:NEXT:POKE S+
    1,100:POKE S+5,120:POKE S+15,255
645 POKE S+24,15:POKE S+4,129
650 RETURN
660 FOR T=1 TO 300:NEXT:PRINT"J"
665 REM CLR SCRN
670 S=54272:FOR W=0 TO 24:POKE S+W,0:NEXT:POKE S
    +1,100:POKE S+5,50:POKE S+15,10
675 POKE S+24,15:FOR T=1 TO 255:POKE S+1,T:POKE
    S+4,33:POKE S+4,18:NEXT
680 FOR T=1 TO 200:NEXT:POKE V+21,0:POKE 53281,
    8:POKE 53280,8
690 PRINT"#####A PRISONER HAS ESCAPED!!!":
    FOR T=1 TO 2000:NEXT
695 REM DOWN CURR BLACK
698 PRINT"#####YOU SCORED#####";SC

```

```

845 REM DOWN CRSR RED WHITE
850 FOR T=1 TO 2000:NEXT:PRINT"DO YOU WANT
  ANOTHER GAME?"
851 REM DOWN CRSR YELLOW
855 GET A$:IF A#<>"Y" THEN 865
860 RUN
865 IF A#<>"N" THEN 855
870 POKE 53281,9:POKE 53280,9:PRINT"*****":END
875 REM BROWN CLR DOWN
900 IF SC=AW+1000 THEN PL=PL+1:PRINT"*****
  LEVEL#  ":PL:GOTO 910
901 REM HOME DOWN WHITE
905 RETURN
910 AW=SC:L=L+2:LV=LV+4:FOR T=1 TO 1000:NEXT
915 PRINT"*****":RETURN
916 REM HOME DOWN
920 RETURN
1000 JV=PEEK(197):IF JV=60 THEN FR=10
1010 IF JV=37 THEN Y=Y-8
1011 IF JV=36 THEN Y=Y+8
1012 IF JV=12 THEN X=X-8
1013 IF JV=23 THEN X=X+8
1015 GOTO 38

```

ASTRO-BLASTER

The space freighter lanes around your space station are dangerously cluttered with a group of large asteroids. You have been sent out in your trusty mining ship to blast the asteroids out of the way.

However, there is a complication. An alien culture has decided that these asteroids are worthy of preservation and has sent a patrol ship to protect them. This alien is a pacifist as well as a conservationist so he will not fire at your ship as you try to remove the rocks. However, he will not hesitate to ram you to protect the asteroids. The alien has a radar tracking device which will home in on your ship. You are a little lucky as this device doesn't work very well when you are over on the right hand side of the screen.

As well as avoiding the alien's attempts to ram you, you must also avoid crashing into the asteroids or the border of the screen. Any of these mishaps will cause your ship to explode. You will find that your missiles have no effect on the alien ship or on any asteroid the alien ship lands on.

The "I" key moves your ship up the screen and the "M" key moves it down screen. The "J" key moves the ship left and the "L" key moves it to the right. The "U" key can be used as a temporary pause button to help you regain your breath! This key arrangement may seem a little difficult at first but, you'll discover, it becomes a very good way to control four-directional movement. The "F" key fires your missiles.

The movement of your ship is continuous. When you press a key, your ship keeps moving in that direction until you press another key. This makes the game very difficult and it requires a lot of playing to master.

A final word of warning. Your ship is a mining ship, not a space fighter. It wasn't designed to carry heavy weapons, so when you fire your missiles watch out for the recoil!

After the title has been displayed, the screen will turn black with a brown border. The four asteroids will be POKEd onto the screen followed by the alien ship near the top left hand corner of the screen and your ship in the middle of the screen but nearer to the

bottom. The time taken will be displayed in the top left hand corner of the screen.

Astro-blaster was written by Daniel Quinn.

```
1 REM *** ASTRO-BLASTER ***
2 X$="JOHN SMITH":BEST$="50":RT=0.25
10 RESTORE:FOR P=1 TO 25:PRINT:NEXT:PRINT"☐":
   REM HOME
11 POKE 53281,0:POKE 53280,9
12 PRINT"██████████":TAB(12):"███ASTRO-BLASTER███"
13 REM DOWN CRSR BLUE REV ON REV OFF
20 V=53248
21 POKE V+21,0
22 POKE V,0:POKE V+1,195
23 POKE V+14,0:POKE V+15,150
24 POKE V+2,0:POKE V+3,170
25 POKE V+16,0
26 MX=V+16
27 C=2040
30 GOSUB 9030
40 PRINT"☐":REM CLR SCRN
41 ZX=RND(-TI)
42 GOSUB 1500
50 X=150:Y=195:S=15:AS=4
60 EX=25:EY=50:S2=RT*S
80 POKE V+41,7:POKE V+42,2:POKE V+43,8:POKE
   V+44,9
95 POKE V+21,127
96 POKE V+39,7:POKE V+40,7
97 POKE 197,0
99 TI$="000000"
100 P=PEEK(197):IF P=4 THEN PRINT"☐":GOTO 10:
   REM CLR SCRN
101 PRINT"TIME-->"RIGHT$(TI$,3)"    ":REM HOME
102 MSB=PEEK(V+16)
110 IF P=64 THEN P=L+32
115 P=P-32
120 IF P=-11 AND M=0 THEN 122
121 GOTO 125
122 X1=X:Y1=Y:M=1:P=36
```

```

123 IF PEEK(V+16)=1 THEN POKE V+16,3
124 GOTO 115
125 IF P<10RP>37 THEN 190
130 ON P GOTO 140,150,180,160,170
140 Y=Y-S:GOTO 180
150 X=X-S:GOTO 180
160 Y=Y+S:GOTO 180
170 X=X+S
180 L=P
190 IF M=1 THEN Y1=Y1-S:POKE V+2,X1:POKE V+3,Y1
    :IF Y1<30 THEN M=0:X1=0:Y1=0
200 IF X>EX THEN EX=EX+S2
210 IF X<EX THEN EX=EX-S2
220 IF Y<EY THEN EY=EY-S2
230 IF Y>EY THEN EY=EY+S2
475 IF X>255 THEN POKEV+16,1:X=X-255
476 IF X<0 AND PEEK(V+16)=1 THEN POKE V+16,0:
    X=255+X
477 IF X<0 AND PEEK(V+16)=3 THEN POKE V+16,0:
    X=255+X
480 POKE V,X: POKE V+1,Y
490 POKE V+12,EX:POKE V+13,EY
790 W=PEEK(V+30)
795 IF W=65 THEN GOTO 10900
810 IF W=210 THEN POKE 2044,199:POKE 2044,195:
    POKE 2044,13
820 IF W=6 OR W=10 OR W=18 OR W=34 THEN 840
830 GOTO 900
840 N=W-2
850 IF N=4 THEN P0=4:SP=2
855 IF N=8 THEN P0=6:SP=3
860 IF N=16 THEN P0=8:SP=4
870 IF N=32 THEN P0=10:SP=5
880 POKE 2040+SP,199:POKE V+39+SP,10
885 FOR Z=1 TO 100:NEXT
890 POKE 2040+SP,195:POKE V+39+SP,9
891 FOR Z=1 TO 100:NEXT:POKE V+39+SP,7:FOR Z=1
    TO 100:NEXT
892 POKE 2040+SP,13
893 X1=0:Y1=100:M=0
894 POKE V+2,X1:POKE V+3,Y1:AS=AS-0.5
895 IF PEEK(C+2)+PEEK(C+3)+PEEK(C+4)+PEEK(C+5)

```



```

896 PRINT"█"X"          "Y"          ":REM HOME
900 IF X<24AND PEEK(V+16)=0 OR Y<50 OR Y>225 OR
    X>65 AND PEEK(V+16)=1 THEN10900
920 IF W-1=4 OR W-1=8 OR W-1=16 OR W-1=32 THEN
    10900
1000 GOTO 100
1500 T=1:T2=2
1501 POKE V+29,12:POKE V+23,20
1510 FOR Q=1 TO 4:B=0
1511 T=T+1:T2=T2+2
1520 X(Q)=INT(RND(T)*231)+24
1530 Y(Q)=INT(RND(T)*120)+50
1535 IF X(Q)>130 AND X(Q)<170 THEN 1520
1536 IF B=1 THEN 1545
1540 POKE V+21,PEEK(V+21)+(2↑T)
1545 POKE V+T2,X(Q):POKE V+T2+1,Y(Q)
1551 IF PEEK(V+30)> 0 THEN B=1:GOTO 1520
1560 IF PEEK(V+30)=0 THEN NEXT
1600 RETURN
5000 END
6000 REM *** AFTER STAGE ***
6001 FOR Z=1 TO 500:NEXT
6005 POKE V+21,0
6006 POKE 53281,7:PRINT"█":REM BROWN
6010 PRINT"██████████"
6015 REM CLR SCRN DOWN CRSR
6020 PRINT"██████████████████████JELL DONE!!█"
6025 REM RIGHT CRSR REV ON
6030 PRINT:PRINT"█ YOU HAVE CLEARED THIS
    SECTOR OF          ASTEROIDS."
6035 REM BLUE
6040 PRINT:PRINT"█          YOU DID IT IN THE TIME
    OF "
6045 REM BLACK
6050 PRINT"██████████████████"SC$" SECONDS"
6055 REM RIGHT CRSR
6060 PRINT"██████████PREVIOUS BEST:"BEST$"
    SECONDS":PRINT"██████████ACHIEVED BY"X$
6062 REM RIGHT CRSR RED
6065 FOR Z=1 TO 500:NEXT
6070 IF VAL(BEST$)<=VAL(SC$) THEN 6200

```

```

6080 BEST$=SC$
6090 PRINT:PRINT:PRINT:PRINT"#####YOU HAVE
    ACHIEVED BEST TIME"
6095 REM BLUE RIGHT CRSR
6100 INPUT" WHAT IS YOUR NAME,COMMANDER";X$
6900 PRINT"JWELL, VICTORIOUS COMMANDER"
6905 REM CLR SCRNR
6910 PRINT" WHAT DO YOU WISH NOW?"
6920 PRINT" 1)....REPLAY SAME DIFFICULTY "
6925 PRINT" 2)....REPLAY AT EASIER LEVEL "
6930 PRINT" 3)....ADVANCE TO HARDER LEVEL"
6940 PRINT" 4)....END PROGRAM"
7001 PRINT:INPUT"                YOUR ANSWER"
    ;Z:IF Z=4 THEN END
7002 IF Z=1 THEN GOTO 10
7003 IF Z=3 THEN RT=RT+0.25:GOTO10
7004 IF Z=2 THEN RT=RT-0.25:GOTO10
9030 POKE C,192:REM YOUR SHIP.SP(0)
9040 POKE C+1,193:REM MISSILE...SP(1)
9050 POKE C+2,194:REM ASTEROID...SP(2)
9060 POKE C+3,194:REM ASTEROID...SP(3)
9070 POKE C+4,194:REM ASTEROID...SP(4)
9080 POKE C+5,194:REM ASTEROID...SP(5)
9090 POKE C+6,198:REM ALIEN.....SP(6)
9100 FOR A=0 TO 62:READ D:POKE12288+A,D:NEXT
9110 FOR A=0 TO 62:READ D:POKE12736+A,D:NEXT
9120 FOR A=0 TO 62:READ D:POKE12480+A,D:NEXT
9130 FOR A=0 TO 62:READ D:POKE12416+A,D:NEXT
9140 FOR A=0 TO 62:READ D:POKE12352+A,D:NEXT
9145 FOR A=0 TO 62:POKE832+A,0:NEXT
9146 FOR A=0 TO 62:READ D:POKE12672+A,D:NEXT
9200 RETURN
10000 REM *** DATA FOR YOUR SPACE SHIP **
10005 DATA 0,24,0,0,24,0,4,24,32,4,60,32,4,126,
    32,4,231,32,5,195,160
10010 DATA 7,129,224,7,189,224,15,255,240,23,
    189,232,55,255,236,47,189,244
10020 DATA 119,195,238,255,255,255,7,231,224,7,
    255,224,0,195,0,0,0,0,0,0,0
10025 DATA 0,0,0
10030 REM *** DATA FOR EXPLOSION PT 1 ***
10040 DATA 0,0,0,0,0,0,0,130,0,4,64,0,10,12,160

```

```

,72,200,0,23,161,0,6,128,0
10041 DATA 0,0,0,1,32,0,4,64,0
10050 DATA 16,5,0,9,16,0,70,0,32,20,34,16,2,128
,0,16,70,32,1,32,0,9,0,0
10060 DATA 0,64,0,64,0,0
10070 REM *** DATA FOR EXPLOSION PT 2 ***
10080 DATA 128,16,1,0,0,0,1,0,64,0,4,0,0,0,0,9,
32,0,0,0,72,5,0,0
10090 DATA 3,2,0,135,169,1,1,0,0,0,48,16,2,17,
32,0,0,8,8,2,0,0,8,16
10100 DATA 1,0,0,0,2,64,0,0,0,0,0,0,128,8,1
10110 REM *** DATA FOR ASTEROID ***
10120 DATA 0,0,0,0,0,0,0,0,0,0,64,0,0,232,0,5,
188,0,3,250,128,7,181,192
10130 DATA 15,255,192,15,239,160,23,255,192,62,
254,224,25,247,192
10140 DATA 52,189,192,27,118,128,22,59,0,15,92,
0,5,224,0,1,128,0,0,0,0
10150 DATA 0,0,0
10160 REM *** DATA FOR MISSILE ***
10170 DATA 0,24,0,0,24,0,0,60,0,0,24,0,0,24,0,
0,60,0,0,126,0,0,24,0
10180 DATA 0,24,0,0,24,0,0,60,0,0,0,0,0,20,0,0,
42,0,0,20,0,0,66,0
10190 DATA 0,40,0,0,20,0,0,1,0,0,64,0,0,8,0
10200 REM *** DATA FOR ALIEN ***
10210 DATA 0,24,0,0,0,0,0,24,0,0,36,0,0,60,0,0,
102,0,1,219,128,1,0,128
10220 DATA 3,129,192,7,255,224,2,219,64,7,24,
224,12,255,48,24,24,24
10230 DATA 48,24,12,240,62,15,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0
10908 POKE V+29,PEEK(V+29)+1
10909 POKE V+23,PEEK(V+23)+1:XC=50
10910 FOR A=1 TO 5
10912 FOR Z=1 TO XC:NEXT
10913 POKE 2040,199:POKE V+39,2
10914 FOR Z=1 TO XC:NEXT
10915 POKE 2040,195:POKE V+39,9
10916 FOR Z=1 TO XC:NEXT

```

```

10917 POKE 2040,195:POKE V+39,7
10918 FOR Z=1 TO XC:NEXT
10919 NEXT:POKE V+21,0:GOTO 20010
20010 PRINT"7      WELL DONE. YOU MANAGED TO LET
      YOUR ONLY DEFENCE SHIP GET "
20012 REM CLR SCREEN
20015 PRINT"  DESTROYED. MAKING EASY THE
      CONQUEST OF  OUR PLANET"
20040 PRINT:PRINT:PRINT
20050 PRINT"WOULD YOU WISH TO PUT THE TIME
      MACHINE BACK AND TRY AGAIN?";
20055 REM YELLOW
20060 GET ANS$:IF ANS$="Y" THEN 10
20070 IF ANS$="N"THEN END
20080 GOTO 20060

```

COSMIC JOURNEY

Are you ready to brave the dangers of a Cosmic Journey? You must fly your space craft across the screen to the exit point on the other side. The game has a wide range of speed options and many different levels. Each time you successfully complete one screen you move onto the next level. Each new level will be harder than the last.

Your final score in the game will relate to the speed you chose and the number of levels you completed. You get bonus points for flying at faster speeds. The first screen you traverse is empty but after that you will have to negotiate waves of asteroids and, in higher levels, a space barrier will block your path.

Move your space ship to the left and the right with the “J” key and the “K” key respectively. The “D” key will slow your ship and the “S” key will speed up your flight.

At the beginning of the game you will be asked which speed you wish to try. There is no limit to the number you may enter but level three is easy, level five will provide a challenge and level ten is virtually impossible.

When the journey begins, the screen will clear to black and your space ship will appear at the bottom. An exit point will be visible at the top of the screen. There will be a great blast of sound as your ship takes off. Then you must steer the ship across the screen and through the exit point. The screen will clear and a new level will be printed up.

The second level has a scattering of obstacles and these grow more numerous as the game progresses. If your flying skills aren't up to the level required, your ship will disappear in a very impressive explosion.

Cosmic Journey was written by Daniel Quinn.

```

1 REM *** COSMIC-JOURNEY ***
2 V=53248:X=125:Y=225:BD=9:BG=0:L=1
3 POKE V+32,BD:POKE V+33,BG:PRINT CHR$(147)
4 FOR Z=1 TO 10:PRINT:NEXT
5 PRINT TAB(14);"COSMIC-JOURNEY":REM BLUE
6 PRINT TAB(14);"SPEED ";;INPUT SP:REM
  ORANGE
7 S=54272:POKE S+24,15
8 POKE S+5,192:POKE S+6,192
10 PRINT "J":REM CLR SCRNM
110 FOR I=0 TO 62:READ D:POKE 12288+I,D:NEXT
120 FOR I=0 TO 62:READ D:POKE 12352+I,D:NEXT
295 POKE V+21,1:POKE V+39,14
300 POKE V,X:POKE V+1,Y
403 K=INT(RND(1)*20)+10:FOR A=1024 TO 1024+K:
  POKE S+A,BD:POKE A,160:NEXT
404 FOR N=1027+K TO 1063:POKE S+N,BD:POKE N,160
  :NEXT:POKE V+39,8
499 POKE V+30,0:POKE V+31,0:POKE 2040,192
500 P=PEEK(197):POKE S+24,15:Y=Y-SP
510 IF P=34 THEN X=X-SP
520 IF P=37 THEN X=X+SP
521 IF P=18 THEN Y=Y+SP/2
522 IF P=13 THEN Y=Y-SP*2
525 IF X<1 THEN 890
530 POKES+1,X/2:POKE S,Y:POKE S+4,17
531 POKE S+1,Y:POKE S,X/2:POKE S+4,129
890 IF X>255 THEN X=X-255:POKE V+16,1
891 IF X<0 THEN POKE V+16,0:X=255+X
900 POKE V+0,X:POKE V+1,Y
910 IF PEEK(V+31) AND 1=1 THEN 5000:REM GOTO
  5000 ** IF CRASH **
920 IF Y>33 THEN 990
930 Y=229:POKE V+21,0:L=L+1:OK=K
935 PRINT "J":SC=SC+L*SP+10:REM CLR SCRNM
936 FOR Z=1 TO 7:PRINT:NEXT
937 PRINT TAB(13);"SCORE":SC:FOR A=1 TO 500
  :NEXT
940 PRINT"J":FOR A=1984 TO 1984+OK:POKE 54272+A
  ,BD:REM CLR SCRNM

```

```

942 POKE A,160:NEXT:FOR N=1987+OK TO 2023
945 POKE S+N,BD:POKE N,160:NEXT
950 K=INT(RND(1)*20)+10:PRINT"SCORE":REM HOME
    DOWN CRSR
954 FOR A=1024 TO 1024+K:POKE S+A,BD:POKE A,
    160:NEXT
955 FOR N=1027+K TO 1063:POKE 54272+N,BD:POKE
    N,160:NEXT:Y=229
956 IF L<6 THEN L1=L:GOTO 969
957 QW=INT((K+OK)/2)
958 FOR A=1384 TO 1384+QW:POKE A,160:POKE
    54272+A,BD:NEXT
960 FOR A=QW+1389 TO 1423: POKE A,160:POKE
    54272+A,BD:NEXT
965 L1=SP
969 POKE V,X:POKE V+1,Y:POKE V+21,1:POKE V+39,8
970 PRINT"SCORE":FOR Z=1 TO 120:RN=INT
    (RND(1)*10):REM HOME DOWN CRSR
975 IF RN<L1/2 THEN PRINT"3":GOTO 983:REM
    ORANGE
982 PRINT" ";
983 NEXT
984 FOR Z=255 TO 1 STEP -1:POKE S+1,Z:POKE
    S,256-Z:NEXT
985 FOR Z=1 TO 10:FOR Z2=1 TO 2
986 POKE S+1,Z:POKE S,Z2:POKE S+1,S2:POKE S,Z:
    NEXT:NEXT
990 POKE S+1,Y:POKE S,Y
999 POKE 54273,X:POKE 54272,Y
1000 GOTO 500
5000 REM *** IF CRASH ***
5020 POKE 2040,193
5030 FOR Z=1 TO 30:POKE V+23,1:POKE V+23,0:POKE
    V+29,1:POKE V+29,0
5040 POKE V+23,1:POKE V+29,1:POKE V+23,0:POKE
    V+29,0:NEXT
5050 POKE 54273,0:POKE 54272,0
5100 PRINT"3":REM CLEAR SCREEN
5110 PRINT TAB(15):" GAME OVER "
5120 PRINT"SCORE":REM 8 CRSR DOWN
5130 PRINT TAB(14):"FINAL SCORE:";SC

```

```

5135 PRINT TAB(14);"FINAL LEVEL:";L
5140 POKE V+21,0
5200 END

10000 REM *** DATA FOR SPACE SHIP ***
10010 DATA 0,16,0,0,56,0,0,56,0,0,56
10020 DATA 0,0,56,0,0,56,0,0,56,0,0,56,0,0,124,
      0,0,124,0,0,56,0,0,56,0,0,56
10030 DATA 0,0,56,0,0,56,0,0,56,0,0,124,0,0,124,
      0,0,0,0,0,0,0,0,0,0
10035 REM *** DATA FOR EXPLOSION ***
10040 DATA 32,0,64,0,0,0,1,2,0,0,18,80,34,0,129,
      0,41,16,1,50,0,8,152,64,17
10050 DATA 165,96,8,110,64,129,255,72,6,174,32,
      0,254,4,16,236,0,0,25
10060 DATA 112,0,24,64,3,0,35,17,45,0,0,68,0,0,
      0,6,129,0,32

```


SPACE RACE

This short program creates a very challenging game, complete with levels of difficulty. You are the pilot of a V-wing space racer. You must race against the clock, flying a slalom course through blue star. Each time you fly through a star gate your score is incremented. After 30 gates have been passed or flown through the game ends and you are given your score.

There are four levels of difficulty in this game. Level four is for beginners and level one is strictly for experts only.

After the title appears on the screen, you will be asked which level of difficulty you wish to try. When you have entered this, the screen will clear and be set to black. Your V-shaped space racer will be positioned in the middle of the screen on the top row. The game is then on. The "M" key moves your racer to the right and the "Z" key moves it to the left. Each time you press a key the racer moves one space in the required direction.

The blue star gates will rush toward you and you must fly through the blue area to score points. After 30 gates have passed, you will be shown your score. The game also has a high score feature so that you can work at improving your skills.

Lines 600 to 650 display the game title and ask you for your level of difficulty. Line 640 accepts your input and then we RETURN to the main game.

Line 15 sets the starting position of your space craft across the screen (the X axis). Line 50 multiplies your selected level of difficulty by four. The screen is then cleared and set to black.

Line 120 POKes your V-racer onto the screen. This is done by adding the value of X to 1024, the value of the top left hand corner location of the screen memory. The rest of the line POKes a value into the corresponding location in the color memory.

Line 130 adds one to our counter and then checks to see if the counter is greater than a number picked at random between four and eleven plus the value of A. If it is, we then go to the routine in lines 200 to 220. This routine sets our counter back to zero and

then gives variable D a value picked at random between ten and thirty. The variable E is incremented by one. E is used to count the number of star gates passed. When E equals thirty one, the game ends.

Lines 210 and 215 PRINT the star gate on the screen using graphic characters from the "N" and "M" keys on the keyboard and several reversed spaces. The variable D, set in line 200, is then used to TAB the star gate across the screen.

We go back to the main routine where line 135 POKes a blank space into the current location of your space racer. This makes sure that two space racers don't appear on the screen at the same time when the space racer is moving across the screen.

Line 140 GETs your input from the keyboard. Lines 150 and 160 check the keyboard input to see if one of the direction keys has been pressed. If so, one is added to, or subtracted from, the value of X depending on which key has been pressed. Line 165 then POKes your space racer onto the screen again, in its new position if you have pressed a key, or in its original position if you haven't.

Line 170 uses the level of difficulty to provide a delay loop, which is, of course, longer for the easier levels. This loop has the effect of slowing down the program.

Line 175 calls up the routine in lines 300 and 320. This routine uses PEEK to check the values of the locations on either side of your racer. If it finds a reverse space, (value 160), this means that you have flown through a star gate and your score is incremented.

Line 180 loops back to the start of the flight routine for another pass. Line 110, at the start of this routine, makes sure that a new line is printed each time. This keeps the star gates moving smoothly up the screen. The star gates appear to jump when they arrive on the screen. This is caused by three lines being printed instead of the usual one.

```

5 REM SPACE RACE
10 PRINT "□":GOSUB 600:REM CLR SCRN
15 X=20
50 A=A#4
60 PRINT "□"
70 POKE 53280,0:POKE 53281,0
100 REM ROUTINE
110 PRINT
120 POKE 1024+X,22:POKE 55296+X,7
130 C=C+1:IF C>INT(RND(1)*7)+4+A THEN 200
135 POKE 1024+X,96
140 GET A$
150 IF A$="Z" THEN X=X-1
160 IF A$="M" THEN X=X+1
165 POKE 1024+X,22:POKE 55296+X,7
170 FOR T=1 TO A:NEXT T
175 GOSUB 300
180 GOTO 100
200 C=0:D=INT(RND(1)*20)+10
205 E=E+1:IF E=31 THEN PRINT "GAME OVER":FOR
    T=1 TO 2000:NEXT T:GOSUB 500
206 REM REV ON HOME
210 PRINT TAB(D) "  \  /":REM WHITE
212 PRINT TAB(D) "  3  4":REM REV ON BLUE
    THREE SPACES REV OFF
215 PRINT TAB(D) "  /  \":REM WHITE
220 GOTO 100
300 IF PEEK(1024+X-1)=160 OR PEEK(1024+X+1)=160
    THEN SC=SC+10*E
320 RETURN
500 POKE 53280,2:POKE 53281,3:PRINT "□"
510 PRINT TAB(10) "YOUR SCORE WAS";SC
520 IF SC>HS THEN HS=SC
530 PRINT TAB(4) "THE BEST SCORE SO FAR IS"
    ;HS
540 FOR TD=1 TO 1000:NEXT
550 PRINT TAB(4) "DO YOU WANT ANOTHER GAME
    (Y/N)";
560 INPUT A$
570 IF A$="Y" THEN SC=0:E=0:GOTO 10
580 END

```

```
600 PRINT "I":POKE 53280,2:POKE 53281,6
610 PRINT TAB(14) "XXXXXXXXXXSPACE RACE"
620 FOR TD=1 TO 1500:NEXT TD
630 PRINT TAB(7) "XCHOOSE FLIGHT SPEED (1-4)";
640 INPUT A
650 RETURN
```

GHOSTIE

In this game you are the fearless warrior charged with the difficult task of keeping the nasty ghosties out of the grave-yard. You must prevent the red ghost from gaining safety amongst the headstones. If you kill *any* of the ghosts you *stop* the red ghost. You will score more points for *killing* the red ghost but he is harder to fight than the others. You can move your warrior all around the screen by using either the keyboard or the joystick. To destroy a ghost you must press the fire button (or the space bar in the keyboard version) just before you engage the ghosts in combat.

The first listing of the game is the joystick version and the second listing is for the keyboard. When playing the keyboard version the "F" key moves your warrior up the screen and the "V" key moves him down the screen. The "J" and "K" keys are for left and right movement respectively. To kill a ghost, use the space bar.

The game can end in any one of three ways. The ghosts can kill you, the red ghost can get past you into the graveyard or if you stray into the top left hand corner of the screen (where the ghosts appear) you will be zapped.

When you RUN the game, after choosing one of four levels of difficulty, the screen clears to black and the game is on.

The graveyard is PRINTed at the bottom of the screen, your warrior is somewhere in the middle and the ghosts begin to enter from the top left hand corner of the screen.

The initialisation routine is contained in lines 2 to 22. The sprites are set up by lines 13 and 195 and 197. These sprites are used for the warrior and the two ghosts. The warrior is a particularly good sprite and looks great on the screen.

Line 126 reads the joystick input (or the keyboard input). The score is worked out in line 27 using the value of TI\$, the internal clock.

Lines 28 to 35 use the value from the input to change the X and Y value which are used to POKE the warrior sprite onto the screen.

Lines 36 to 39 are to make sure that the warrior stays on the screen and are valuable as safeguards. It is always a good idea to put such safeguards into a program which uses moving graphics. This prevents the values in the POKE statements leaving the screen memory locations and getting in with other memory locations.

The lines 41 to 55 check to see if either of the ghosts is in a direct line with the warrior and if so, we then go down to the routine in lines 300 to 449 where the ghosts are moved in to follow our man. If the ghosts aren't in line with him, lines 59 to 69 move the red ghost around the screen at random.

Line 70 looks to see if the man has wandered into the top left hand corner of the screen and, if he has, 'zaps' him. This prevents a player from sitting there and waiting for the ghosts to appear.

The sprites are POKEd into the screen memory locations by lines 100 and 105. Lines 119 and 120 check for a collision between sprites. Lines 130 and 150 look to see if the fire button has been pressed (or if space bar has been hit) before the collision. Line 160 PRINTs the current score and then goes back to the start of the game loop.

Lines 200 to 275 contain the DATA for the sprites. This must be typed in very carefully.

The lines 450 to 452 test to see if the fire button was pressed during a collision. If it was, lines 460 to 480 PRINT up the message that the ghost was destroyed and then start off two more ghosts. If it wasn't, lines 500 to 539 PRINT the message that the man was beaten and tell you your score. You are also given the option of another game.

Lines 40 to 545 re-initialise the program for a new game. Lines 547 to 550 end the whole program if you wish to stop.

If the red ghost reaches the graveyard, lines 1000 and 1100 tell you that you were beaten and call up the end routine.

The opening sequence which asks for the level of difficulty and then clears the screen and PRINTs the graveyard, can be found in lines 3000 to 5200. Ghostie was written by William Leader.

```

1 REM *** GHOSTIE JOYSTICK VERSION ***
2 PRINT"D":AG=0:POKE 53281,0:POKE 53280,0:REM
  CLD:GOTO 1
3 RESTORE
10 V=53248:POKE V+21,28
12 POKE 2042,194:POKE 2043,195:POKE 2044,196
13 FOR N=12416 TO 12478:READ Q:POKE N,Q:NEXT:
  GOTO 195
15 L=25:R=60:GH=100:POKE V+43,14:POKE V+42,2:
  POKE V+41,10
16 I=INT(200*RN(1))+40):K=INT(200*RN(1))+8):
  X=100:Y=100
18 IF K>X-17 AND K<X+17 THEN 15
19 IF K<L-17 AND K<X+17 THEN 15
20 GOTO 3000
22 POKE 53281,0:GOTO 100
26 JV=PEEK(56320):FR=JV AND 16:JV=15-(JV AND 15)
27 SC=SC+VAL(TI$)-(AG*2)
28 IF JV=1 THEN Y=Y-CM
29 IF JV=2 THEN Y=Y+CM
30 IF JV=4 THEN X=X-CM
31 IF JV=8 THEN X=X+CM
32 IF JV=5 THEN Y=Y-CM:X=X-CM
33 IF JV=6 THEN Y=Y+CM:X=X-CM
34 IF JV=9 THEN Y=Y-CM:X=X+CM
35 IF JV=10 THEN Y=Y+CM:X=X+CM
36 IF X>254 THEN X=X-15
37 IF X<1 THEN X=X+15
38 IF Y>250 THEN Y=Y-15
39 IF Y<1 THEN Y=Y+15
41 IF X<L-10 AND X<L+10 THEN 300
44 IF Y<R-10 AND Y<R+10 THEN 400
51 IF X>I-10 AND X<I+10 THEN 415
55 IF Y>K-10 AND Y<K+10 THEN 420
59 G=INT(2*RN(1))+1
60 IF G=1 THEN L=L-INT(PM*RN(1))+5
61 IF G=2 THEN L=L+INT(PM*RN(1))+5:
62 IF L>200 THEN L=L-10
63 IF L<5 THEN L=L+10
64 G=INT(2*RN(1))+1
65 IF G=1 THEN R=R-INT((PM-2)*RN(1))+5:
66 IF G=2 THEN R=R+INT((PM+2)*RN(1))+5

```

```

67 IF R>210 AND (L>140 AND L<268) THEN 1000
68 IF R>230 THEN R=230
69 IF R<5 THEN R=R+10
70 IF X<44 AND Y<45 THEN PRINT "YOU ARE ZAPPED":
    FOR FG=1 TO 2000: NEXT: POKE V+21,0: GOTO 527
100 POKE V+4,X: POKE V+6,L: POKE V+8,I
105 POKE V+5,Y: POKE V+7,R: POKE V+9,K
119 IF (L>X-10 AND L<X+10) AND (R>Y-15 AND
    R<Y+15) THEN SC=SC+500: GOTO 449
120 IF (I>X-10 AND I<X+10) AND (K>Y-15 AND
    K<Y+15) THEN 449
130 IF FR=16 THEN 160
150 IF FR<>16 THEN AG=AG+1: IF AG=5 THEN JT=2
153 IF L>200 THEN L=190
160 PRINT "M": SC: GOTO 26: REM HOME CRSR
193 POKE V+21,4
195 FOR S2=12480 TO 12542: READ Q2: POKE S2,Q2:
    NEXT
197 FOR S3=12544 TO 12606: READ Q3: POKE S3,Q3:
    NEXT: GOTO 15
200 DATA 0,0,16,0,24,8,0,36,4,0,36,2,0,24,3,0,
    24,7,0,230,14,1,129,252
210 DATA 3,129,248,15,129,240,14,66,0,28,66,0,56
    ,60,0,112,62,0,224,85,0
220 DATA 64,5,0,1,5,0,2,5,0,5,5,0,10,5,0,10,7,
    128
250 DATA 0,16,2,0,40,6,0,84,14,0,186,30,0,84,62,
    0,40,126,0,56,254,0,223,230
255 DATA 1,215,226,255,239,192,255,95,128,127,
    224,0,96,222,0,97,156,0
260 DATA 97,172,0,66,216,0,2,184,0,7,88,0,12,176
    ,0,31,240,0,31,240,0
265 DATA 0,16,2,0,40,6,0,84,14,0,186,30,0,84,62,
    0,40,126,0,56,254,0,223,230
270 DATA 1,215,226,255,239,192,255,95,128,127,
    224,0,96,222,0,97,156,0
275 DATA 97,172,0,66,216,0,2,184,0,7,88,0,12,176

```



```

,0,31,240,0,31,240,0
300 IF Y<R THEN R=R-EM:GOTO 51
310 R=R+EM:GOTO 51
400 IF X<L THEN L=L-EM:GOTO 51
410 L=L+EM:GOTO 51
415 IF Y<K THEN K=K-EM:GOTO 59
416 K=K+EM:GOTO 100
420 IF X<I THEN I=I-EM:GOTO 59
421 I=I+15:GOTO 100
449 IF JT=2 THEN 500
450 IF RE=2 THEN RE=0:GOTO 26
451 IF FR<>16 THEN PRINT"YOU GOT HIM":REM
    CLR SCRNM PURPLE LT BLUE
452 IF FR=16 THEN 500
460 POKE V+21,0
465 FOR U=1 TO 1000:NEXT:PRINT" "
470 PRINT TAB(12)"BUT ANOTHER TWO":SC=SC+1500
472 REM PURPLE LT BLUE
475 FOR U=1 TO 1000:NEXT:AG=0:FR=16:PRINT" "
477 L=25:R=60:RE=2
478 I=INT(200*VRND(1)+40):LET K=I
479 IF I>X-20AND I<X+20 THEN 477
480 A$="Y":GOTO 5000
500 FOR RY=1 TO 10
505 POKE 53281,10:FOR T=1 TO 100:NEXT
510 POKE 53281,1:FOR T=1 TO 100:NEXT
520 NEXT:POKE V+21,0
525 PRINT"HE GOT YOU":FOR C=1 TO 1000:NEXT:
    GOTO 530:REM CLR SCRNM LT RED
527 X=100:Y=100
530 POKE 53281,0:PRINT" ":PRINT"YOUR FINAL
    SCORE IS":SC:REM CLR SCRNM LT RED
531 IF SC>20000 THEN PRINT"A FANTASTIC SCORE"
    :GOTO 535
532 IF SC>10000 THEN PRINT"A GREAT SCORE":
    GOTO 535
533 IF SC>5000 THEN PRINT"A COOL SCORE":
    GOTO 535
534 IF SC<1000 THEN PRINT"A ROTTEN SCORE"
535 PRINT"DO YOU WANT ANOTHER GAME":REM DOWN
    CRSR YELLOW

```

```

536 GET A$:IF A$<>"Y" THEN 538
537 GOTO 540
538 IF A$<>"N" THEN 536
539 GOTO 547
540 PRINT "I":SC=0:JT=0:AG=0
541 POKE V+21,0
542 I=INT(200*RND(1)+40):LET K=I:RE=2
543 L=25:P=60:IF IX=20 AND ICX+20 THEN 541
544 FOR NM=1 TO 2000:NEXT:TI$="000000":SC=0:
    GOTO 5000
547 POKE V+21,0
549 PRINT "BYE BYE":FOR T=1 TO 2000:NEXT:
    PRINT "I":POKE 53281,6
550 REM DOWN CRSR LT BLUE CLR SCRN
551 END
1000 POKE V+7,230
1100 PRINT "THE BEAT YOU I":FORE=1 TO 2000:NEXT:
    SC=SC-1000:POKE V+21,0:PRINT "I":GOTO 530
1105 REM USE ABBREVIATION FOR PRINT OR LINE
    WON'T FIT
3000 PRINT "WHAT DIFFICULTY?"
3005 PRINT "(V=VERY HARD/H=HARD/A=AVERAGE/
    E=EASY)"
3010 INPUT RT$
3100 IF RT$="E" THEN EM=10:PM=5:CM=14
3200 IF RT$="A" THEN EM=12:PM=8:CM=12
3210 IF RT$="H" THEN EM=15:PM=10:CM=10
3220 IF RT$="V" THEN EM=20:PM=15:CM=12
3250 PRINT "I":PRINT:POKE 53281,0
3400 PRINT "I":FR=16:RE=2:FOR TRT=1 TO 100:NEXT
5000 FOR R=1 TO 21:PRINT:NEXT
5100 PRINT TAB(15)"  O  O  O":REM GRAY 2
5110 PRINT TAB(15)"II  II  II"
5120 PRINT TAB(15)"  ##  ##  ##"
5130 PRINT TAB(15)"  ###  ###  ###":POKE V+21,28:IF
    A$="Y" THEN 100
5200 GOTO 22

```

```

1 REM *** GHOSTIE KEYBOARD VERSION ***
2 PRINT "G":AG=0:POKE 53281,0:POKE 53280,0:REM
  CLR SCRN
3 RESTORE
10 V=53248:POKE V+21,28
12 POKE 2042,194:POKE 2043,195:POKE 2044,196
13 FOR N=12416 TO 12478:READ Q:POKE N,Q:NEXT:
  GOTO 195
15 L=25:R=60:GH=100:POKE V+43,14:POKE V+42,2:
  POKE V+41,10
16 I=INT(200*RND(1)+40):K=INT(200*RND(1)+8):
  X=100:Y=100
18 IF K>X-17 AND K<X+17 THEN 15
19 IF K>L-17 AND K<L+17 THEN 15
20 GOTO 3000
22 POKE 53281,0:GOTO 100
26 JV=PEEK(197):FR=16:IF JV=60 THEN FR=10
27 SC=SC+VAL(TI$)-(AG*2)
28 IF JV=21 THEN Y=Y-CM
29 IF JV=31 THEN Y=Y+CM
30 IF JV=34 THEN X=X-CM
31 IF JV=37 THEN X=X+CM
36 IF X>254 THEN X=X-15
37 IF X<1 THEN X=X+15
38 IF Y>250 THEN Y=Y-15
39 IF Y<1 THEN Y=Y+15
41 IF X>L-10 AND X<L+10 THEN 300
44 IF Y>R-10 AND Y<R+10 THEN 400
51 IF X>I-10 AND X<I+10 THEN 415
55 IF Y>K-10 AND Y<K+10 THEN 420
59 G=INT(2*RND(1))+1
60 IF G=1 THEN L=L-INT(PM*RND(1))+5
61 IF G=2 THEN L=L+INT(PM*RND(1))+5:
62 IF L>200 THEN L=L-10
63 IF L<5 THEN L=L+10
64 G=INT(2*RND(1))+1
65 IF G=1 THEN R=R-INT((PM-2)*RND(1))+5:
66 IF G=2 THEN R=R+INT((PM+2)*RND(1))+5
67 IF R>210 AND (L>140 AND L<200) THEN 1000
68 IF R>230 THEN R=230
69 IF R<5 THEN R=R+10

```

```

70 IFX<44ANDY<45THENPRINT"YOU ARE ZAPPED":
FORFGH=1TO2000:NEXT:POKEV+21,0:GOTO527
100 POKE V+4,X:POKE V+6,L:POKE V+8,I
105 POKE V+5,Y:POKE V+7,R:POKE V+9,K
110 IF (L>X-10 AND L<X+10) AND (R>Y-15 AND
RCY+15) THEN SC=SC+500:GOTO 449
120 IF (I>X-10 AND I<X+10) AND (K>Y-15 AND
KCX+15) THEN 449
130 IF FR=16 THEN 160
150 IF FRC>15 THEN AG=AG+1:IF AG=5 THEN JT=2
153 IF L>200 THEN L=190
160 PRINT" ";SC:GOTO 26:REM HOME CRSR
193 POKE V+21,4
195 FOR S2=12480 TO 12542:READ Q2:POKE S2,Q2:
NEXT
197 FOR S3=12544 TO 12606:READ Q3:POKE S3,Q3:
NEXT:GOTO 15
200 DATA 0,0,16,0,24,8,0,36,4,0,36,2,0,24,3,0,
24,7,0,230,14,1,129,252
210 DATA 3,129,248,15,129,240,14,66,0,28,66,0,
56,60,0,112,62,0,224,85,0
220 DATA 64,5,0,1,5,0,2,5,0,5,5,0,10,5,0,10,7,
128
250 DATA 0,16,2,0,40,6,0,84,14,0,186,30,0,84,62,
0,40,126,0,56,254,0,223,230
255 DATA 1,215,226,255,239,192,255,95,128,127,
224,0,96,222,0,97,156,0
260 DATA 97,172,0,66,216,0,2,184,0,7,88,0,12,
176,0,31,240,0,31,240,0
265 DATA 0,16,2,0,40,6,0,84,14,0,186,30,0,84,62,
0,40,126,0,56,254,0,223,230
270 DATA 1,215,226,255,239,192,255,95,128,127,
224,0,96,222,0,97,156,0
275 DATA 97,172,0,66,216,0,2,184,0,7,88,0,12,
176,0,31,240,0,31,240,0
300 IF Y<R THEN R=R-EM:GOTO 51
310 R=R+EM:GOTO 51
400 IF X<L THEN L=L-EM:GOTO 51

```

```

410 L=L+EM:GOTO 51
415 IF Y<K THEN K=K-EM:GOTO 59
416 K=K+EM:GOTO 100
420 IF X<I THEN I=I-EM:GOTO 59
421 I=I+15:GOTO 100
449 IF JT=2 THEN 500
450 IF RE=2 THEN RE=0:GOTO 26
451 IF FR<16 THEN PRINT"YYOU GOT HIMY":REM CLR
    SCRN PURPLE LT BLUE
452 IF FR=16 THEN 500
460 POKE V+21,0
465 FOR U=1 TO 1000:NEXT:PRINT"Y"
470 PRINT TAB(12)"YBUT ANOTHER TWOY":SC=SC+1500
472 REM PURPLE LT BLUE
475 FOR U=1 TO 1000:NEXT:AG=0:FR=16:PRINT"Y"
477 L=25:R=60:RE=2
478 I=INT(200*RND(1)+40):LET K=I
479 IF I>X-20AND I<X+20 THEN 477
480 A$="Y":GOTO 5000
500 FOR RV=1 TO 10
505 POKE 53281,10:FOR T=1 TO 100:NEXT
510 POKE 53281,1:FOR T=1 TO 100:NEXT
520 NEXT:POKE V+21,0
525 PRINT"YHE GOT YOUY":FOR C=1 TO 1000:NEXT:
    GOTO 530:REM CLR SCRN LT RED
527 X=100:Y=100
530 POKE 53281,0:PRINT"Y":PRINT"YOUR FINAL
    SCORE ISY":SC:REM CLR SCRN LT RED
531 IF SC>20000 THEN PRINT"A FANTASTIC SCORE"
    :GOTO 535
532 IF SC>10000 THEN PRINT"A GREAT SCORE"
    :GOTO 535
533 IF SC>5000 THEN PRINT"A COOL SCORE"
    :GOTO 535
534 IF SC<1000 THEN PRINT"A ROTTEN SCORE"
535 PRINT"YWANT YOU WANT ANOTHER GAME":REM DOWN
    CRSR YELLOW
536 GET A$:IF A$<>"Y" THEN 538
537 GOTO 540
538 IF A$<>"N" THEN 536

```

```

539 GOTO 547
540 PRINT "J":SC=0:JT=0:AG=0
541 POKE V+21,0
542 I=INT(200*RND(1)+40):LET K=I:RE=2
543 L=25:R=60:IF I>X-20 AND I<X+20 THEN 541
545 FOR NM=1 TO 2000:NEXT:TI$="000000":SC=0:
    GOTO 5000
547 POKE V+21,0
549 PRINT"YOUBYE BYE":FOR T=1 TO 2000:NEXT:
    PRINT "JJ":POKE 53281,6
550 REM DOWN CURSR LT BLUE CLR SCRN
551 END
1000 POKE V+7,230
1100 PRINT "THE BEAT YOUJ":FORE=1TO2000:NEXT:
    SC=SC-1000:POKEV+21,0:PRINT "J":GOTO530
1105 REM USE ABBREVIATION FOR PRINT OR LINE
    WON'T FIT
3000 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXWHAT DIFFICULTY?"
3005 PRINT"(V=VERY HARD/H=HARD/A=AVERAGE/
    E=EASY)"
3010 INPUT RT$
3100 IF RT$="E" THEN EM=10:PM=5:CM=14
3200 IF RT$="A" THEN EM=12:PM=8:CM=12
3210 IF RT$="H" THEN EM=15:PM=10:CM=10
3220 IF RT$="V" THEN EM=20:PM=15:CM=12
3250 PRINT "J":PRINT:POKE 53281,0
3400 PRINT "J":FR=16:RE=2:FOR TRT=1 TO 100:NEXT
5000 FOR R=1 TO 21:PRINT:NEXT
5100 PRINT TAB(15)"  O  O  O":REM GRAY 2
5110 PRINT TAB(15)"II  II  II"
5120 PRINT TAB(15)"===  ===  ==="
5130 PRINT TAB(15)"###  ###  ###":POKE V+21,28:IF
    A$="Y" THEN 100
5200 GOTO 22

```

ASTROSMASH

Arcade action and excitement is provided by this game written by William Leader. The game uses sprite graphics, the sound generating capabilities of the Commodore 64 and nine levels of difficulty to provide a very challenging game.

Your aim is to protect your star base from a shower of asteroids. Your space ship must destroy the asteroids with its blaster to prevent them from falling onto the base below. If three asteroids get past you, or if you collide with an asteroid, the game is over. After you have destroyed ten asteroids the game automatically advances to the next level of difficulty.

We have provided two listings of Astromash, the first one is a joystick operated version with the joystick plugged into port number two. The joystick moves your ship left and right and the fire button fires the blaster. The second version is keyboard operated and in this game the "Z" key moves the ship to the left, the "L" key moves the ship to the right and the space bar fires your blaster.

After the colourful opening sequence, you will be asked to INPUT a level of difficulty. Level one is the easiest and level nine the most difficult. After you have made your selection, the screen will clear and be set to black. A background of stars appears with your spaceship at the bottom of the screen. After a short delay, asteroids will begin falling toward you. From then on it is up to you to save your star base from the asteroid shower.

Lines 1 to 24 initialise the program, setting up the starting values for all the variables. Lines 11 to 24 set up the sprite registers and read in the data to build the sprites.

Line 28 reads the INPUT from the joystick or keyboard. Then lines 30 and 31 use the values and the space ship is moved and tilted accordingly. Variable PO controls the tilt of the ship. Value 1 is tilt left, value 2 is upright and value 3 is tilt right. The score is printed by line 33.

Lines 50 to 55 get the relevant memory locations for the ship, depending on which way it is tilted. Line 60 moves the asteroids. The variable SP is the speed at which they move. This is

determined by the level of difficulty. Line 65 and line 66 make sure that the asteroids stay on the screen.

Lines 70 and 74 check to see if the asteroid has reached the bottom of the screen and, if so, increment the variable HIT. If HIT equals three then we go to the routine which ends the game.

Line 80 checks to see if any of the sprites have collided, then lines 82 and 83 test to find what has collided with what. Line 85 counts the number of destroyed asteroids and advances the program to the next level after ten hits have been scored.

Lines 90 to 100 POKE the sprites into the correct screen locations. Line 115 moves your shot if one is currently on the screen. Line 118 checks to see if the fire button has been pressed, and if it has, and there is no current shot on the screen, starts a shot on its way. Line 120 sets the whole loop in motion once again.

Lines 130 to 275 contain the DATA for the sprites. Take extra care to enter these lines accurately. A mistake in these DATA statements will cause problems with the sprites on the screen.

Lines 300 to 310 move your shot on the screen. FD is the variable for fire direction. Value 1 moves your shot to the left, value 2 moves it straight up the screen, and value 3 moves your shot to the right. Collision checks are made in lines 320 to 530.

The explosion for the first asteroid is POKEd onto the screen by lines 550 to 565. The second asteroid's explosion is handled by lines 570 to 585. The explosion of the space ship is controlled by lines 700 to 720.

The end routine is held in lines 750 to 800. The colourful title sequence is in lines 900 to 2220. The sound routines are in lines 3000 to 3220.

Lines 4000 to 4050 are the routine which advances the game to the next level of difficulty after ten asteroids have been hit. The final lines, 4100 to 4200 turn off the sound channels.


```

1 REM *** ASTROSMASH ***
2 REM *** JOYSTICK VERSION ***
5 POKE 53281,10:POKE 53280,10:PRINT "Q":REM CLR
  SCRN
6 X2=200*RND(1)+50:Y2=40*RND(1):X3=200*RND(1)
  +50:Y2=40*RND(1)
10 V=53248:SC=0
11 PO=2:POKE 2044,196:POKE 2045,197
12 POKE2040,192:POKE2041,193:POKE2042,194:X=155
  :Y=185:POKE2043,195:POKE2046,198
13 FOR S=12544 TO 12606:READ Q:POKE S,Q:NEXT
14 RESTORE:FOR S=12608 TO 12670:READ Q:POKE S,Q
  :NEXT
15 FOR S=12288 TO 12350:READ Q:POKE S,Q:NEXT
17 FOR S=12352 TO 12414:READ Q:POKE S,Q:NEXT
20 FOR S=12416 TO 12478:READ Q:POKE S,Q:NEXT
21 FOR S=12480 TO 12542:READ Q:POKE S,Q:NEXT
22 FOR S=12672 TO 12734:READ Q:POKE S,Q:NEXT
24 POKE V+39,7:POKE V+40,7:POKE V+41,7:POKE
  V+43,9:POKE V+44,8:GOTO 900
25 FOR SO=1 TO 100
28 JV=PEEK(56320):FR=JV AND 16:JV=15-(JV AND 15)
30 IF JV=4 THEN PO=PO-1:IF PO<1 THEN PO=1:X=X-5
31 IF JV=8 THEN PO=PO+1:IF PO>3 THEN PO=3:X=X+5
33 PRINT"Q":SC:REM HOME CRSR
50 IF PO=1 THEN XN=0:W=54
53 IF PO=2 THEN XN=2
55 IF PO=3 THEN XN=4
60 Y2=Y2+SP:Y3=Y3+SP
65 IF X> 250 THEN X=250
66 IF X<20 THEN X=20
70 IF Y2>250 THEN HIT=HIT+1:Y2=0
71 IF Y3>250 THEN HIT=HIT+1:Y3=0
74 IF HIT>2 THEN 750
80 IF PEEK(V+30) AND 4<>4 THEN 90
82 IF X2<X+10 AND X2>X-10 AND Y2<Y+10 AND
  Y2>Y-10 THEN 700
83 IF X3<X+10 AND X3>X-10 AND Y3<Y+10 AND
  Y3>Y-10 THEN 700
85 IF TY>9 THEN LV=LV+1:TY=0:GOSUB 4000
90 POKE V,0:POKE V+1,0:POKE V+3,255:POKE V+5,0

```

```

100 POKE V+(XN),X:POKE V+6,SX:POKE V+8,X2:POKE
    V+10,X3
110 POKE V+(XN+1),Y:POKE V+7,SY:POKE V+9,Y2:POKE
    V+11,Y3
115 IF SH=100 THEN 300
118 IF FRC>16 THEN FD=PO:SH=100: SX=X:SY=Y:POKE
    V+21,PEEK(V+21) OR 8:GOSUB 3000
120 NEXT:GOTO 25
130 DATA 0,56,0,0,126,0,3,255,248,7,199,252,4,
    138,252,97,138,254,27,195,252
132 DATA 31,231,252,63,255,248,63,255,240,62,225
    ,192,28,225,252,56,243,252
135 DATA 56,242,60,62,254,60,127,252,124,127,248
    ,224,15,255,192,7,255,128
140 DATA 0,255,0,0,0,0
150 DATA 0,0,0,0,0,0,14,0,0,9,128,0,10,192,0,5,
    224,0,5,144,0
155 DATA 3,15,255,1,4,3,0,130,13,0,193,16,0,160,
    224,0,137,32
160 DATA 0,142,32,0,153,144,0,142,112,0,152,240,
    0,176,240,0,160,0,0,192,0
165 DATA 0,224,0
200 DATA 0,24,0,0,36,0,0,90,0,0,90,0,0,90,0,0,90
    ,0,0,90,0,0,66,0,0,66,0
210 DATA 0,66,0,0,66,0,0,195,0,1,66,128,2,90,64,
    4,90,32,8,90,16,16,153,8
220 DATA 32,189,4,79,189,242,80,126,10,96,60,6
230 DATA 0,0,0,0,0,28,0,0,100,0,0,212,0,0,232,31
    ,255,120,24,4,48
235 DATA 22,8,32,3,16,64,1,160,128,0,193,128,0,
    130,128,1,34,128
240 DATA 3,72,128,3,136,128,7,156,128,7,196,128,
    0,2,128,0,2,128,0,1,128,0,2,128
250 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,60,0,0,60,0
255 DATA 0,60,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

```

```

260 DATA 0,16,0,0,56,3,0,56,15,0,56,30,0,124,60
    ,128,128,120,255,255,248
265 DATA 255,255,248,127,255,240,63,255,240,31,
    255,224,15,255,224
270 DATA 7,255,224,15,255,128,31,252,0,62,124,0,
    56,62,0,48,31,128
275 DATA 96,15,192,0,3,192,0,1,192
300 IF FD=1 THEN SX=SX-20:SY=SY-20:GOTO 320
305 IF FD=2 THEN SY=SY-20:GOTO 320
310 IF FD=3 THEN SX=SX+20:SY=SY-20:GOTO 320
320 IF PEEK(V+30) AND 4=4 THEN 500
322 IF SY<400RSX>2600RSX<25THENSH=0:POKEV+21,
    PEEK(V+21)AND(255-8):SY=0:SX=0
325 IF SX>230 AND FD=3 THEN SX=270:GOTO 322
330 GOTO 120
500 IF SX>X2-10ANDSX<X2+10ANDSY>Y2-10ANDSY<Y2+
    10ANDSH=100THEN550
510 IF SX>X3-10ANDSX<X3+10ANDSY>Y3-10ANDSY<Y3+10
    ANDSH=100THEN570
530 GOTO 322
550 POKEV+6,SX:POKEV+7,SY:POKEV+21,PEEK(V+21)OR
    64:POKEV+12,SX:POKEV+13,SY
555 POKE V+8,0:POKE V+9,0
560 X2=200*VRND(1)+50:Y2=0:GOSUB3100:FORT=1TO500
    :NEXT:SC=SC+20:SH=0:SY=0
565 POKE V+21,PEEK(V+21)AND(255-72):TY=TY+1:
    GOTO 100
570 POKEV+6,SX:POKEV+7,SY:POKEV+21,PEEK(V+21)OR
    64:POKEV+12,SX:POKEV+13,SY
575 POKE V+10,0:POKE V+11,0
580 X3=200*VRND(1)+50:Y3=0:GOSUB3100:FORT=1TO500
    :NEXT:SC=SC+20:SH=0:SY=0
585 POKE V+21,PEEK(V+21)AND(255-72):TY=TY+1:
    GOTO 100
700 POKE V+21,PEEK(V+21) OR 64:POKE V+12,X:POKE
    V+13,Y:GOSUB 3100
710 FOR T=1 TO 20:IF T/2=INT(T/2) THEN POKE
    V+45,1:GOTO 720

```

```

715 POKE V+45,0
720 FOR O=1 TO 50:NEXT: NEXT
750 POKE V+21,0:POKE 53280,10:POKE 53281,13:
  PRINT"???"
751 REM CLR SCRN RED
755 PRINT"#####YOU FAILED!!!!":REM DOWN CRSR
757 PRINT"#####LEVEL      ":LV:REM GREEN
760 PRINT"#####BUT YOU SCORED":SC:REM BROWN
765 GOSUB 4100:PRINT"#####DO YOU WANT ANOTHER
  GO?????":REM BLUE
770 GET A$:IF A#<>"Y" THEN 780
775 RESTORE:RUN
780 IF A#<>"N" THEN 770
800 END
900 S=54272:FOR L=0 TO 24:POKE S,L:NEXT
905 POKE S+5,9:POKE S+6,0:POKE S+24,15:HF=25:
  LF=177
1000 GOSUB 3200
1005 PRINT"#####
1007 REM CLEAR SCREEN BLUE
1010 PRINT"#####
1020 PRINT"#####
1030 PRINT"#####
1040 PRINT"#####
1050 PRINT"#####
1060 PRINT"#####
1070 PRINT"#####
1080 PRINT
1085 PRINT"#####
1090 PRINT"#####
1095 PRINT"#####
2000 PRINT"#####
2010 PRINT"#####
2015 PRINT"#####
2020 PRINT"#####
2025 PRINT"#####
2030 PRINT"#####
2100 FOR T=1 TO 50 :IF T/2=INT(T/2) THEN POKE
  53281,1:POKE 53280,10:GOTO 2120
2110 POKE 53281,10:POKE 53280,1
2120 POKE S+1,HF:POKE S,LF:POKE S+4,33:FOR M=1
  TO 50:NEXT:HF=HF+1:LF=LF+1

```

```

2125 POKE S+4,32:NEXT
2130 POKE 53280,0:IU=30
2200 PRINT"XXXXXXXXXXXXWHAT LEVEL?(1-9)
2205 REM CLR SCRN DOWN CRSR
2210 GET A$:IF A$="" THEN 2300
2215 SP=(VAL(A$))/2:HIT=0:POKE 53280,0:LV=SP*2
2217 PRINT"□":POKE 53281,0:FOR T=1 TO 25:POKE
    1024+INT(1000*RND(1)),46:NEXT
2220 POKE V+21,55:GOTO 25
2300 IU=IU+1:IF(IU/35)=INT(IU/35) THEN POKE S+4,
    20:GOSUB 3210
2310 GOTO 2210
3000 S=54272
3005 FOR L=0 TO 24:POKE S+L,0:NEXT
3010 POKE S+1,10 :POKE S+5,14 :POKE S+15,18:POKE
    S+24,15:POKE S+4,129
3020 RETURN
3100 S=54272
3105 FOR L=0 TO 24:POKE S+L,0:NEXT
3110 POKE S+1,5 :POKE S+5,140:POKE S+15,2 :POKE
    S+24,15:POKE S+4,129
3120 RETURN
3200 S=54272:FOR L=0 TO 24:POKE S+L,0:NEXT
3205 POKE S+1,130:POKE S+5,9:POKE S+15,30:POKE
    S+24,15:RETURN
3210 POKE S+4,21:FOR P=1 TO 100:NEXT:RETURN
4000 POKE 53281,1
4004 PRINT"XXXXXXXXXXXXLEVEL    ";LV:
    SP=SP+0.5
4005 REM CLR SCRN DOWN CRSR BLUE RIGHT CRSR
4010 S=54272:FOR L=0 TO 24:POKE S+L,0:NEXT
4015 POKES+5,9:POKES+6,0:POKES+24,15:LF=100:HF=
    150:FORT=1TO100:LF=LF-1:HF=HF-1
4020 POKE S+1,HF:POKE S,LF:POKE S+4,33:FOR O=1
    TO 10:NEXT:POKE S+4,32:NEXT
4030 PRINT"□":POKE 53281,0:FOR T=1 TO 25:POKE
    1024+INT(1000*RND(1)),46:NEXT
4040 X2=200*RND(1)+50:Y2=10:X3=200*RND(1)+50:
    Y3=0:POKE 53280,0
4050 PRINT"□":RETURN:REM LT BLUE

```

```

4100 S=54272:FOR L=0 TO 24:POKE S+L,0:NEXT
4105 POKES+5,9:POKES+6,0:POKES+24,15:POKES+4,33
4120 HF=200:LF=250:POKES+4,33:FORT=1T050:HF=HF-1
      :LF=LF-1:POKES+1,HF:POKES,LF
4125 POKE S+4,33:FOR I=1 TO 10:NEXT:POKE S+4,32
      :NEXT
4130 HF=30:LF=40:POKE S+4,33:FORT=1T010:HF=HF+1
      :LF=LF+1:POKE S+1,HF:POKE S,LF
4135 POKE S+4,33:FOR I=1 TO 10:NEXT:POKE S+4,32
      :NEXT
4140 HF=30:LF=40:POKE S+4,33:FORT=1T010:HF=HF-1
      :LF=LF-1:POKE S+1,HF:POKE S,LF
4145 POKE S+4,33:FOR I=1 TO 10:NEXT:POKE S+4,32
      :NEXT
4150 HF=30:LF=40:POKE S+4,33:FORT=1T020:HF=HF+1
      :LF=LF+1:POKE S+1,HF:POKE S,LF
4155 POKE S+4,33:FOR I=1 TO 10:NEXT:POKE S+4,32
      :NEXT
4200 RETURN

```

```

1 REM *** ASTROSMASH ***
2 REM *** KEYBOARD VERSION ***
5 POKE 53281,10:POKE 53280,10:PRINT "J":REM CLR
  SCRN
6 X2=200*RND(1)+50:Y2=40*RND(1):X3=200*RND(1)
  +50:Y2=40*RND(1)
10 V=53248:SC=0
11 PO=2:POKE 2044,196:POKE 2045,197
12 POKE2040,192:POKE2041,193:POKE2042,194:X=155
      :Y=185:POKE2043,195:POKE2046,198
13 FOR S=12544 TO 12606:READ Q:POKE S,Q:NEXT
14 RESTORE:FOR S=12608 TO 12670:READ Q:POKE S,Q
      :NEXT

```

```

15 FOR S=12288 TO 12350:READ Q:POKE S,Q:NEXT
17 FOR S=12352 TO 12414:READ Q:POKE S,Q:NEXT
20 FOR S=12416 TO 12478:READ Q:POKE S,Q:NEXT
21 FOR S=12480 TO 12542:READ Q:POKE S,Q:NEXT
22 FOR S=12672 TO 12734:READ Q:POKE S,Q:NEXT
24 POKE V+39,7:POKE V+40,7:POKE V+41,7:POKE V+43
    ,9:POKE V+44,8:GOTO 900
25 FOR S0=1 TO 100
28 JV=PEEK(197):IF JV=60 THEN FR=10:GOTO 30
29 FR=16
30 IF JV=12 THEN P0=P0-1:IF P0<1 THEN P0=1:X=X-5
31 IF JV=42 THEN P0=P0+1:IF P0>3 THEN P0=3:X=X+5
33 PRINT"筭":S0:REM HOME CRSR
50 IF P0=1 THEN XN=0:W=54
53 IF P0=2 THEN XN=2
55 IF P0=3 THEN XN=4
60 Y2=Y2+8P:Y3=Y3+8P
65 IF X>250 THEN X=250
66 IF X<20 THEN X=20
70 IF Y2>250 THEN HIT=HIT+1:Y2=0
71 IF Y3>250 THEN HIT=HIT+1:Y3=0
74 IF HIT>2 THEN 750
80 IF PEEK(V+30) AND 4<>4 THEN 90
82 IF X2<X+10 AND X2>X-10 AND Y2<Y+10 AND
    Y2>Y-10 THEN 700
83 IF X3<X+10 AND X3>X-10 AND Y3<Y+10 AND
    Y3>Y-10 THEN 700
85 IF TY>9 THEN LV=LV+1:TY=0:GOSUB 4000
90 POKE V,0:POKE V+1,0:POKE V+3,255:POKE V+5,0
100 POKE V+(XN),X:POKE V+6,SX:POKE V+8,X2:POKE
    V+10,X3
110 POKE V+(XN+1),Y:POKE V+7,SY:POKE V+9,Y2:POKE
    V+11,Y3
115 IF SH=100 THEN 300
118 IF FR<>16 THEN FD=P0:SH=100:SX=X:SY=Y:POKE
    V+21,PEEK(V+21) OR 8:GOSUB 3000
120 NEXT:GOTO 25
130 DATA 0,56,0,0,126,0,3,255,248,7,199,252,4,
    138,252,97,138,254,27,195,252
132 DATA 31,231,252,63,255,248,63,255,240,62,225
    ,192,28,225,252,56,243,252

```

```

135 DATA 56,242,60,62,254,60,127,252,124,127,248
    ,224,15,255,192,7,255,128
140 DATA 0,255,0,0,0,0
150 DATA 0,0,0,0,0,0,14,0,0,9,128,0,10,192,0,5,
    ,224,0,5,144,0
155 DATA 3,15,255,1,4,3,0,130,13,0,193,16,0,160,
    ,224,0,137,32
160 DATA 0,142,32,0,153,144,0,142,112,0,152,240,
    ,0,176,240,0,160,0,0,192,0
165 DATA 0,224,0
200 DATA 0,24,0,0,36,0,0,90,0,0,90,0,0,90,0,0,90
    ,0,0,90,0,0,66,0,0,66,0
210 DATA 0,66,0,0,66,0,0,195,0,1,66,128,2,90,64,
    ,4,90,32,0,90,16,16,153,8
220 DATA 32,189,4,79,189,242,80,126,10,96,60,6
230 DATA 0,0,0,0,0,28,0,0,100,0,0,212,0,0,232,31
    ,255,120,24,4,48
235 DATA 22,0,32,3,16,64,1,160,128,0,193,128,0,
    ,130,128,1,34,128
240 DATA 3,72,128,3,136,128,7,156,128,7,196,128,
    ,0,2,128,0,2,128,0,1,128,0,2,128
250 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,60,0,0,60,0
255 DATA 0,60,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0,0,0,0,0,0,0,0
260 DATA 0,16,0,0,56,3,0,56,15,0,56,30,0,124,60,
    ,128,128,120,255,255,248
265 DATA 255,255,248,127,255,240,63,255,240,31,
    ,255,224,15,255,224
270 DATA 7,255,224,15,255,128,31,252,0,62,124,0
    ,56,62,0,48,31,128
275 DATA 96,15,192,0,3,192,0,1,192
300 IF FD=1 THEN SX=SX-20:SY=SY-20:GOTO 320
305 IF FD=2 THEN SY=SY-20:GOTO 320
310 IF FD=3 THEN SX=SX+20:SY=SY-20:GOTO 320
320 IF PEEK(V+30) AND 4=4 THEN 500
322 IF SY<400ORSX>260ORSX<25 THEN SH=0:POKEV+21,
    PEEK(V+21)AND(255-8):SY=0: SX=0

```



```

325 IF SX>230 AND FD=0 THEN SX=270:GOTO 322
330 GOTO 120
500 IFSX>X2-10ANDSX<X2+10ANDSY>Y2-10ANDSY<Y2+
    10ANDSH=100THEN550
510 IFSX>X3-10ANDSX<X3+10ANDSY>Y3-10ANDSY<Y3+
    10ANDSH=100THEN570
530 GOTO 322
550 POKEV+6,SX:POKEV+7,SY:POKEV+21,PEEK(V+21)OR
    64:POKEV+12,SX:POKEV+13,SY
555 POKE V+8,0:POKE V+9,0
560 X2=200*VRND(1)+50:Y2=0:GOSUB3100:FORT=1TO500
    :NEXT:SC=SC+20:SH=0:SY=0
565 POKE V+21,PEEK(V+21)AND(255-72):TY=TY+1:GOTO
    100
570 POKEV+6,SX:POKEV+7,SY:POKEV+21,PEEK(V+21)OR
    64:POKEV+12,SX:POKEV+13,SY
575 POKE V+10,0:POKE V+11,0
580 X3=200*VRND(1)+50:Y3=0:GOSUB3100:FORT=1TO500
    :NEXT:SC=SC+20:SH=0:SY=0
585 POKE V+21,PEEK(V+21)AND(255-72):TY=TY+1:
    GOTO 100
700 POKE V+21,PEEK(V+21) OR 64:POKE V+12,X:POKE
    V+13,Y:GOSUB 3100
710 FOR T=1 TO 20:IF T/2=INT(T/2) THEN POKE
    V+45,1:GOTO 720
715 POKE V+45,0
720 FOR O=1 TO 50:NEXT:NEXT
750 POKE V+21,0:POKE 53280,13:POKE 53281,13:
    PRINT"34"
751 REM CLR SCRN RED
755 PRINT"YOUYOU FAILED!!!" :REM DOWN CRSR
757 PRINT"LEVELLEVEL" :LV:REM GREEN
760 PRINT"BUTBUT YOU SCORED":SC:REM BROWN
765 GOSUB 4100:PRINT"DODO YOU WANT ANOTHER
    GO?????":REM BLUE
770 GET A$:IF A$<>"Y" THEN 780
775 RESTORE:RUN
780 IF A$<>"N" THEN 770
800 END

```

```

900 S=54272:FOR L=0 TO 24:POKE S,L:NEXT
905 POKE S+5,9:POKE S+6,0:POKE S+24,15:HF=25:
    LF=177
1000 GOSUB 3200
1005 PRINT"
1007 REM CLEAR SCREEN BLUE
1010 PRINT"
1020 PRINT"
1030 PRINT"
1040 PRINT"
1050 PRINT"
1060 PRINT"
1070 PRINT"
1080 PRINT
1085 PRINT"
1090 PRINT"
1095 PRINT"
2000 PRINT"
2010 PRINT"
2015 PRINT"
2020 PRINT"
2025 PRINT"
2030 PRINT"
2100 FOR T=1 TO 50 :IF T/2=INT(T/2) THEN POKE
    53281,1:POKE 53280,10:GOTO 2120
2110 POKE 53281,10:POKE 53280,1
2120 POKE S+1,HF:POKE S,LF:POKE S+4,33:FOR M=1
    TO 50:NEXT:HF=HF+1:LF=LF+1
2125 POKE S+4,32:NEXT
2130 POKE 53280,0:IU=30
2200 PRINT"WHAT LEVEL?(1-9)
2205 REM CLR SCRN DOWN CRSR
2210 GET A$:IF A$="" THEN 2300
2215 SP=(VAL(A$))/2:HIT=0:POKE 53280,0:LV=SP*2
2217 PRINT":POKE 53281,0:FOR T=1 TO 25:POKE
    1024+INT(1000*RND(1)),46:NEXT
2220 POKE V+21,55:GOTO 25
2300 IU=IU+1:IF(IU/35)=INT(IU/35) THEN POKE S+4,
    20:GOSUB 3210
2310 GOTO 2210
3000 S=54272
3005 FOR L=0 TO 24:POKE S+L,0:NEXT

```

```

3010 POKE S+1,10 :POKE S+5,14 :POKE S+15,18:
      POKE S+24,15:POKE S+4,129
3020 RETURN
3100 S=54272
3105 FOR L=0 TO 24:POKE S+L,0:NEXT
3110 POKE S+1,5 :POKE S+5,140:POKE S+15,2 :
      POKE S+24,15:POKE S+4,129
3120 RETURN
3200 S=54272:FOR L=0 TO 24:POKE S+L,0:NEXT
3205 POKE S+1,130:POKE S+5,9:POKE S+15,30:POKE
      S+24,15:RETURN
3210 POKE S+4,21:FOR P=1 TO 100:NEXT:RETURN
4000 POKE 53281,1
4004 PRINT"#####LEVEL      ";LV:SP=SP
      +0.5
4005 REM CLR SCRN DOWN CURR BLUE RIGHT CURR
4010 S=54272:FOR L=0 TO 24:POKE S+L,0:NEXT
4015 POKES+5,9:POKES+6,0:POKES+24,15:LF=100:
      HF=150:FORT=1TO100:LF=LF-1:HF=HF-1
4020 POKE S+1,HF:POKE S,LF:POKE S+4,33:FOR O=1
      TO 10:NEXT:POKE S+4,32:NEXT
4030 PRINT"□":POKE 53281,0:FOR T=1 TO 25:POKE
      1024+INT(1000*VRND(1)),46:NEXT
4040 X2=200*VRND(1)+50:Y2=10:X3=200*VRND(1)+50:
      Y3=0:POKE 53280,0
4050 PRINT"□":RETURN:REM LT BLUE
4100 S=54272:FOR L=0 TO 24:POKE S+L,0:NEXT
4105 POKES+5,9:POKES+6,0:POKES+24,15:POKES+4,33
4110 HF=200:LF=250:POKES+4,33:FORT=1TO50:HF=HF-1
      :LF=LF-1:POKES+1,HF:POKES,LF
4125 POKE S+4,33:FOR I=1 TO 10:NEXT:POKE S+4,32
      :NEXT
4130 HF=30:LF=40:POKE S+4,33:FORT=1TO10:HF=HF+1
      :LF=LF+1:POKE S+1,HF:POKE S,LF
4135 POKE S+4,33:FOR I=1 TO 10:NEXT:POKE S+4,32
      :NEXT
4140 HF=30:LF=40:POKE S+4,33:FORT=1TO10:HF=HF-1
      :LF=LF-1:POKE S+1,HF:POKE S,LF
4145 POKE S+4,33:FOR I=1 TO 10:NEXT:POKE S+4,32
      :NEXT

```

```
4150 HF=30:LF=40:POKE S+4,33:FOR T=1 TO 20:HF=HF+1
      :LF=LF+1:POKE S+1,HF:POKE S,LF
4155 POKE S+4,33:FOR I=1 TO 10:NEXT:POKE S+4,32
      :NEXT
4200 RETURN
```

ADVENTURE AND SIMULATION

Survival

Roman Empire

Battle of Britain

Sunnybrook Farm



Roman Empire

SURVIVAL

This program written by Doug Rolfe is an adventure program with a difference.

You are a passenger on an interstellar flight. The ship has been attacked and ransacked by space pirates. The crew has been killed and the few passengers the ship was carrying have been kidnapped. You are the only person left on board the ship. To survive you must get to the control room, get the ship underway and then get off the ship to safety. Please be very careful, a derelict spaceship holds many dangers for the unwary.

Commands should be entered as two words. Firstly a verb, such as GET or GO, this should be followed by a noun or a direction such as BOOTS or NOSEWARD. You have four choices of directions. Noseward and Tailward, which are self explanatory, Spinward and Antispinward. These last two refer to the direction in which the ship is rotating to produce an artificial gravity.

The screen will display all the information that you need to know as the game proceeds. You will be told what you can see in each of the four directions, and under this you will be informed if there are any objects near you. You will then be asked for your command.

We don't want to make the game too easy for you so we won't give you much more information but there are two more things that you need to know.

Please note that locked doors and locked lockers have to be UNLOCKED. Closed doors need to be OPENED. The OPEN and UNLOCK routines are separate and if you try to OPEN a locked door you may be told there isn't a door there. The program will look for a closed door and not a locked door, if it doesn't find one it will tell you that you have made a mistake. Of course you can only unlock a locked door or locker if you have the right key!

Finally, be very careful what you pick up, once you have something it is yours forever. You can't put it down again! Good luck.

```

10 REM *** SURVIVAL ***
20 PRINT "J":GOSUB 3000:REM CLR SCRIN
25 POKE 53280,2:POKE 53281,10
30 WL$(0)="A WALL":WL$(1)="NOTHING":WL$(2)=
  "THE HULL":WL$(3)="A CLOSED DOOR"
40 WL$(4)="A LOCKED DOOR":WL$(5)="AN OPEN DOOR"
  :WL$(6)="AN AIRLOCK"
50 WL$(7)="A HAND-PRINT CONTROLLED DOOR":WL$(8)
  ="A LOCKED LOCKER"
60 WL$(9)="AN OPEN LOCKER":DR$(1)="NOSEWARD"
  :DR$(2)="TAILWARD"
70 DR$(3)="ANTISPINWARD":DR$(4)="SPINWARD"
80 DIM O$(5,15),D$(5,15),OB$(35)
90 PRINT "J":PRINT "■PLEASE WAIT..."
95 REM BLACK
100 N=14:N2=51:DIM V$(N),N$(N2):X=1:Y=1
110 FOR XX=1 TO N:READ V$(XX):NEXT XX
120 FOR XX=1 TO N2:READ N$(XX):NEXT XX
130 FOR XX=1 TO 5:FOR YY=1 TO 15:READ D$(XX,YY)
  :NEXT YY:NEXT XX
135 GOSUB 4000
140 FOR XX=1 TO 5:FOR YY=1 TO 15:READ O$(XX,YY)
  :NEXT YY:NEXT XX
150 FOR XX=0 TO 34:READ OB$(XX):NEXT XX
160 PRINT "J"
165 REM CLR SCRIN
170 PRINT
180 GOSUB 1040
190 PRINT
200 PRINT "☞YOUR COMMAND PLEASE":INPUT A$:PRINT
  "■":REM BLUE BLACK
205 GOSUB 4000
210 PRINT "000";
215 REM 2 CRSR DOWN
220 F$=LEFT$(A$,3)
230 IF F$="INV" THEN FOR XX=0 TO NP:PRINT PO$(
  XX):NEXT XX:GOTO 190
235 IF LEN(A$)<4 THEN PRINT "I DON'T UNDERSTAND"
  :GOTO 190
240 FOR P=1 TO LEN(A$):P$=MID$(A$,P,1):IF
  P$=" " THEN GOTO 250

```



```

245 NEXT P:PRINT "PLEASE ENTER A TWO WORD
    SENTENCE":GOTO 190
247 REM DOWN CRSR
250 S$=MID$(A$,P+1,3)
260 FOR XX=1 TO N:IF V$(XX)=F$ THEN F=1:V=XX
270 NEXT XX
280 IF F=0 THEN PRINT "NO RECOGNISABLE VERB":
    GOTO 190
290 F=0
300 FOR XX=1 TO N2:IF N$(XX)=S$ THEN F=1:NO=XX
310 NEXT XX
320 IF F=0 THEN PRINT "NO RECOGNISABLE NOUN":
    GOTO 190
330 F=0
340 ON V GOTO 370,470,1160,700,570,1370,1370,
    1790
350 ON (V-8) GOTO 1790,1790,1790,1720,1830,1900
360 PRINT "YOU CAN'T DO THAT":GOTO 190
370 IF MID$(D$(X,Y),NO,1)="1" THEN GOTO 410
380 IF MID$(D$(X,Y),NO,1)="5" THEN GOTO 410
390 IF MID$(D$(X,Y),NO,1)="9" THEN GOTO 410
400 GOSUB 3200:PRINT "THUMP!!!!":GOTO
    170
405 REM RIGHT CRSP
410 ON NO GOTO 430,440,450,460
420 PRINT "THAT DOESN'T MAKE SENSE":GOTO 190
430 Y=Y+1:GOTO 170
440 Y=Y-1:GOTO 170
450 X=X+1:GOTO 170
460 X=X-1:GOTO 170
470 IF K$="NOTHING" THEN PRINT "THERE IS
    NOTHING HERE":GOTO 170
480 FOR P=1 TO LEN(K$):PP$=MID$(S$,P,3):IF PP$
    <>"0" THEN GOTO 490
485 NEXT P:PRINT "I CAN'T SEE ANY SUCH OBJECT"
    :GOTO 170
490 IF NP=7 THEN PRINT "YOU CAN'T CARRY ANY
    MORE":GOTO 190
500 IF K$="BODY" THEN PRINT "THE BODY IS TOO
    HEAVY TO CARRY":GOTO 170
510 IF K$="COMPUTER" THEN PRINT "YOU CAN'T,
    ITS FIXED TO THE WALL":GOTO 170

```

```

520 IF K$="AUTOPILOT INSTRUCTION TAPE" THEN FT=1
530 IF K$="SPACE SUIT" THEN FS=1
540 IF K$="DOOR KEY" THEN DK=1
550 IF K$="LASER CUTTER" OR K$="POWER PACK"
    THEN LF=LF+1
560 IF K$="BACK PACK" THEN FS(5)=1
570 IF K$="LOCKER KEY" THEN LK=1
580 IF K$="SONIC SCREWDRIIVER" THEN T1=1
590 IF K$="LOGIC PROBE" THEN T2=1
600 IF K$="PLIERS" THEN T3=1
610 IF K$="WARHEAD" THEN W1=1:EX=1
620 IF K$="DETONATOR" THEN W2=1:EX=1
630 IF K$="NUCLEAR CHARGE" THEN W3=1:EX=1
640 IF K$="HELMET" THEN FS(1)=1
650 IF K$="SPACE SUIT" THEN FS(2)=1
660 IF K$="GLOVES" THEN FS(3)=1
670 IF K$="BOOTS" THEN FS(4)=1
680 NP=NP+1:PO$(NP)=K$:O$(X,Y)="0"
690 PRINT "YOU HAVE ";RR$;" ";K$:GOTO 170
700 IF NOC>14 THEN GOTO 780
710 PRINT "WHERE";INPUT A$:A$=LEFT$(A$,3)
720 FOR XX=1 TO 4:IF LEFT$(DR$(XX),3)=A$ THEN
    GOTO 740
730 NEXT XX:PRINT "THAT DOESN'T MAKE SENSE":
    GOTO 170
740 IF MID$(D$(X,Y),XX,1)<>"4" THEN PRINT "THERE
    IS NO LOCKED DOOR ";DR$(XX):GOTO 170
741 REM PUT CURSOR BACK INTO LINE BEFORE
    PRESSING RETURN
750 IF DK>1 THEN PRINT "YOU DON'T HAVE THE KEY
    ":GOTO 170
760 PRINT "DOOR UNLOCKED":D$(X,Y)=LEFT$(D$(X,Y)
    ,XX-1)+"3"+RIGHT$(D$(X,Y),4-XX)
770 NW$="3":ON XX GOTO 930,940,950,960
780 IF NOC>15 THEN GOTO 860
790 PRINT "WHERE";INPUT A$:A$=LEFT$(A$,3)
800 FOR XX=1 TO 4:IF LEFT$(DR$(XX),3)=A$ THEN
    GOTO 820
810 NEXT XX:PRINT "THAT DOESN'T MAKE SENSE"
    :GOTO 170
820 IF MID$(D$(X,Y),XX,1)<>"8" THEN PRINT "NO
    LOCKED LOCKER ";DR$(XX):GOTO 170

```

```

830 IF LK<>1 THEN PRINT "YOU DON'T HAVE THE
    KEY!!":GOTO 170
840 D$(X,Y)=LEFT$(D$(X,Y),XX-1)+"9"+RIGHT$(
    (D$(X,Y),4-XX)
850 NW$="9":ON XX GOTO 930,940,950,960
860 PRINT "WHERE":INPUT A$:A$=LEFT$(A$,3)
870 FOR XX=1 TO 4:IF LEFT$(DR$(XX),3)=A$ THEN
    GOTO 890
880 NEXT XX:PRINT "THAT DOESN'T MAKE SENSE!!":
    GOTO 170
890 IF MID$(D$(X,Y),XX,1)<>"6" THEN PRINT"THERE
    IS NO AIRLOCK ";DR$(XX):GOTO170
900 IF RF=1 AND FS=2 AND XC>3 THEN PRINT
    "YOU MADE IT!!":END
910 IF FS<>2 THEN PRINT "YOU EXPLODED IN THE
    VACUUM!!":END
920 PRINT "YOU HAD NO WHERE TO GO AND FLOATED
    TO YOUR DEATH!!":END
930 D$(X,Y+1)=LEFT$(D$(X,Y+1),1)+NW$+RIGHT$(
    (D$(X,Y+1),2):GOTO 170
940 D$(X,Y-1)=NW$+RIGHT$(D$(X,Y-1),3):GOTO 170
950 D$(X+1,Y)=LEFT$(D$(X+1,Y),3)+NW$:GOTO 170
960 D$(X-1,Y)=LEFT$(D$(X-1,Y),2)+NW$+RIGHT$(D$
    (X-1,Y),1)+NW$:GOTO 170
970 PRINT "WHERE":INPUT A$:A$=LEFT$(A$,3)
980 FOR XX=1 TO 4:IF A$=LEFT$(DR$(XX),3)THEN
    GOTO 1000
990 NEXT XX:PRINT "THAT DOESN'T MAKE SENSE!!"
    :GOTO 170
1000 IF MID$(D$(X,Y),XX,1)<>"3" THEN PRINT
    "NO CLOSED DOOR ";DR$(XX):GOTO 170
1010 PRINT "DOOR OPEN":D$(X,Y)=LEFT$(D$(X,Y),
    XX-1)+"5"+RIGHT$(D$(X,Y),4-XX)
1020 NW$="5":ON XX GOTO 930,940,950,960
1030 GOTO 170
1040 FOR TD=1 TO 1000:NEXT TD:PRINT "D":FOR DR=1
    TO 4:PRINT
1045 REM CLR SCRN
1050 PRINT DR$(DR);" YOU SEE ";WL$(VAL(MID$(
    (D$(X,Y),DR,1)))

```

```

1060 NEXT DR
1070 IF VAL(0$(X,Y)) <> 0 THEN K$=0B$(VAL(0$(
    (X,Y))) : GOTO 1140
1080 IF 0$(X,Y) = "0" THEN K$="NOTHING" : RR$=""
    : GOTO 1140
1090 K$=0B$(ASC(0$(X,Y))-55)
1140 PRINT "DO YOU SEE ";K$;" HERE?"
1141 REM YELLOW CRSR DOWN LT BLUE
1150 RETURN
1150 IF 0$(X,Y) <> "P" THEN PRINT "THERE IS NO
    COMPUTER HERE" : GOTO 170
1170 PRINT "DO YOU WISH TO SET A COURSE(Y/N)"
1175 REM CLR SCRN
1180 GET A$: IF A$="" THEN GOTO 1180
1190 IF A$="Y" THEN GOTO 1220
1200 PRINT "I HAVE NO OTHER FUNCTION,"
1210 PRINT "DISCONNECTION PROCEEDS" : GOTO 170
1220 IF FT=1 THEN GOTO 1240
1230 PRINT "IMPOSSIBLE WITHOUT" : PRINT "AUTO
    PILOT COURSE TAPE" : GOTO 170
1240 PRINT "COURSE SET" : PRINT "DO YOU WISH TO"
1245 REM CLR SCRN
1250 PRINT "PROCEED TO DESTINATION? (Y/N)"
1260 GET A$: IF A$="" THEN GOTO 1260
1270 IF A$="N" THEN GOTO 170
1280 IF A$ <> "Y" THEN GOTO 1260
1290 PRINT "HYPERDRIVE IN OPERATION..."
1295 REM CLR SCRN
1300 FOR XX=1 TO 2000 : NEXT XX : PRINT "I" : RF=1 :
    PRINT "PRESENT STATUS:"
1305 REM CLR SCRN DOWN CRSR
1310 PRINT "ORBITING 4TH PLANET OF" : PRINT
    "STAR 'CANOPUS' ."
1320 PRINT "SENTRY CRAFT OF THIS" : PRINT "PLANET
    REQUEST DOCKING"
1330 PRINT "DO YOU WISH TO COMPLY? (Y/N) " :
    PRINT "(ADVISED)"
1335 REM DOWN CRSR
1340 GET A$: IF A$="" THEN GOTO 1340
1350 IF A$="Y" THEN RF=1 : PRINT "DOCKED: PROCEED
    TO AIRLOCK '2'" : GOTO 170

```

```

1360 PRINT "YOU WERE SHOT OUT OF THE SKY!!!!!"
      :END
1370 IF F2=2 THEN PRINT "YOU ALREADY HAVE A
      COMPLETE SUIT":GOTO 190
1380 IF NO=34 THEN GOTO 1440
1390 IF NO=11 OR NO=12 THEN GOTO 1490
1400 IF NO=35 THEN GOTO 1540
1410 IF NO=36 THEN GOTO 1590
1420 IF NO=37 OR NO=35 THEN GOTO 1650
1430 PRINT "THAT DOESN'T MAKE SENSE!!":GOTO 170
1440 IF FS(1)<>1 THEN PRINT "YOU DON'T HAVE A
      HELMET":GOTO 1700
1450 PRINT "YOU ARE NOW WEARING THE HELMET":
      FOR XX=1 TO NP
1460 IF PO$(XX)="HELMET" THEN PO$(XX)=PO$(NP)
      :NP=NP-1:FS(1)=0
1470 NEXT XX
1480 F3=F3+1:GOTO 1700
1490 IF FS(2)<>1 THEN PRINT "YOU DON'T HAVE A
      SPACE SUIT":GOTO 1700
1500 PRINT "YOU ARE NOW WEARING THE SUIT"
      :FOR XX=1 TO NP
1510 IF PO$(XX)="SPACE SUIT" THEN PO$(XX)=PO$(
      (NP):NP=NP-1:FS(2)=0
1520 NEXT XX
1530 F3=F3+1:GOTO 1700
1540 IF FS(3)<>1 THEN PRINT "YOU DON'T HAVE
      GLOVES":GOTO 1700
1550 PRINT "YOU ARE NOW WEARING THE GLOVES":
      FOR XX=1 TO NP
1560 IF PO$(XX)="GLOVES" THEN PO$(XX)=PO$(NP):
      NP=NP-1:FS(3)=0
1570 NEXT XX
1580 F3=F3+1:GOTO 1700
1590 IF FS(4)<>1 THEN PRINT "YOU DON'T HAVE
      BOOTS":GOTO 1700
1600 PRINT "YOU ARE WEARING THE BOOTS":FOR
      XX=1 TO NP
1610 IF PO$(XX)="BOOTS" THEN PO$(XX)=PO$(NP):
      NP=NP-1:FS(4)=1

```

```

1620 NEXT XX
1630 F3=F3+1:GOTO 1700
1640 PRINT "YOU ARE WEARING THE BOOTS":FS2=FS2+
    1:GOTO 170
1650 IF FS(5)<>1 THEN PRINT "YOU DON'T HAVE A
    BACK PACK!":GOTO 1700
1660 PRINT "YOU ARE NOW WEARING THE BACK PACK"
    :FOR XX=1 TO NP
1670 IF PO$(XX)="BACK PACK" THEN PO$(XX)=PO$
    (NP):NP=NP-1:FS(5)=0
1680 NEXT XX
1690 F3=F3+1
1700 IF F3=5 THEN F2=2:NP=NP+1:PO$(NP)="FULL
    SPACE SUIT"
1710 GOTO 170
1720 IF NO<>(16 AND 27) THEN PRINT "YOU CAN'T"
    :GOTO 170
1730 IF LF<>2 THEN PRINT "YOU CAN'T":GOTO 170
1740 FORXX=1TONP:IF PO$(XX)="LASER CUTTER" THEN
    PO$(XX)="CHARGED CUTTER":LF=3
1750 NEXT XX
1760 FOR XX=1 TO NP:IF PO$(XX)="POWER PACK"
    THEN PO$(XX)=PO$(NP):NP=NP-1
1765 NEXT XX
1770 PRINT "YOU HAVE A FULLY CHARGED LASER
    CUTTER    BE CAREFUL!!"
1780 GOTO 170
1790 IF LF=0 THEN PRINT "YOU DON'T HAVE ONE":
    GOTO 170
1800 IF LF<>3 THEN PRINT "NOTHING HAPPENED!"
    :GOTO 170
1810 PRINT "YOU HAVE BURNED A HOLE RIGHT
    THROUGH THEHULL "
1820 PRINT "YOU HAVE DIED IN THE VACUUM":END
1830 IF EX<>1 THEN PRINT "YOU HAVE NOTHING TO
    DETONATE":GOTO 170
1840 IF NO=19 AND W1=1 THEN PRINT "THE WAR HEAD
    HAS DESTROYED THE SHIP"
1842 FOR TD=1 TO 1000:NEXT
1845 PRINT "AND YOU ALONG WITH IT":END
1847 REM DOWN CRSR

```

```

1850 IF N0=18 AND N2=1 THEN PRINT "THE
      DETONATOR HAS BLOWN UP"
1852 FOR TD=1 TO 1000:NEXT
1855 PRINT "IT HAS KILLED YOU":END
1860 IF N3=1 AND (N0=16 OR N0=27) THEN GOTO 1880
1870 PRINT "IMPOSSIBLE!!":GOTO 170
1880 PRINT "ONLY A FOOL WOULD DETONATE A
      NUCLEAR WEAPON INSIDE A SPACESHIP"
1885 FOR TD=1 TO 2500:NEXT
1890 PRINT "BUT YOU DID!!!":END
1895 REM DOWN CRSR
1900 IF N0=14 OR N0=21 THEN GOTO 1920
1910 PRINT "YOU CAN'T":GOTO 170
1920 IF T1<>1 OR T2<>1 OR T3<>1 THEN PRINT
      "YOU DON'T HAVE THE TOOLS":GOTO 170
1930 IF X<>3 OR Y<>12 THEN PRINT "HOW?!!":
      GOTO 170
1940 PRINT "YOU HAVE SUCCESSFULLY DISMANTLED
      THE DOOR"
1950 D$(3,12)="3011":D$(3,13)="1311":GOTO 170
1960 DATA "GO ",GET,CON,UNL,OPE,DON,WEA,LIG,USE
      ,FIR,IGN,ASS,DET,DIS
1970 DATA NOS,TAI,ANT,SPI,NUC,CHA,RIF,PIS,KEY,
      TAP,SPA,SUI,COM,DOO,LOC,CUT
1980 DATA AIR,DET,WAR,TRA,HAN,SCA,LOG,PRO,AUT,
      INS,LAS
1990 DATA MIS,CAS,SCA,KNI,POW,PAC,HEL,GLO,BOD,
      BAC,GRA,HOO,UNI,BIN
2000 DATA BOD,CAB,FOO,WAT,VIT,SOM,SCR,PLI,LOG,
      PRO
2010 DATA 1012,1182,1182,0182,0046,1092,1182,
      0182,1012,1192,1192,8112
2020 DATA 0012,0002,0002
2030 DATA 0211,0088,0088,0088,1014,1109,1108,
      1108,0101,1009,0109,0011
2040 DATA 1011,0110,0000
2050 DATA 0611,1018,1118,4118,0411,1010,4110,
      3410,1310,1100,0110,7011
2060 DATA 1711,1111,2100
2070 DATA 0211,1001,1101,0101,0011,1001,0101,
      0011,0001,0800,0001,0011

```



```
3150 FOR TD=1 TO 250:NEXT
3160 GOSUB 4000
3170 RETURN
3200 POKE VO,15:POKE AD,18:POKE SR,180:POKE
    WF,129
3210 POKE HF,2:POKE LF,37:FOR TD=1 TO 100:NEXT
3220 POKE WF,0:POKE VO,0
3230 GOSUB 4000
3240 RETURN
4000 POKE VO,2:POKE AD,132:POKE SR,196:POKE
    WF,33
4010 POKE HF,INT(3*RND(1)+2)
4020 POKE LF,INT(10*RND(1)+175)
4030 RETURN
```

ROMAN EMPIRE

This program has the potential to make you the most powerful person in the known world, Emperor of the Roman Empire! To be successful you must juggle all your available resources to produce a strong and prosperous Empire. You must balance your trade against the strength of your military, making sure all the time that you are providing enough food for your population.

Each game turn is the equivalent of a year. At the beginning of each year you will be given a Senate report which shows the state of the Empire for the coming year. This report is very important and should be studied carefully. It shows the size of your population, the size of the army and navy, the size of the merchant marine and the amount of labour involved in shipbuilding. It also shows the number of merchant guilds producing goods for export, the amount of uncommitted labour you have to allocate for the year, the size of your gold reserves, how much food you have stockpiled and how much food you will need to feed the people.

The first job you have to do at the start of each turn is to allocate your uncommitted labour for the coming year. This is very important, especially in the early years when the Empire is small and struggling. It is important to get the relationships between the different areas of employment correct. The size of the army decides whether or not you will be able to win wars or put down rebellions in the provinces. The value of your export trade depends on the outcome of these battles.

You must make certain that you have crews to man the number of ships that you are going to build. Each one of your ship building cohorts will build one ship per year. There is no point in having 40 cohorts if you only have 25 crews. Likewise, it is very wasteful to have crews sitting around waiting to have ships built for them. Therefore, you must be sure to allocate enough labour to the navy and merchant navy to crew the ships you are building.

Please note. You can subtract labour from any area just by entering a negative number and you can shuffle labour back and forth to your heart's content until you have no unallocated labour. Once this figure reaches zero the program moves onto the next phase. This may be necessary on the first turn to correct any

imbalances you may start with.

Each one of your guilds of craftsmen produce enough export goods to fill one ship. These trade goods earn your income. The local trading ships provide food for your people.

After all your labour has been allocated, you will be given the opportunity to hire Auxilia units to augment your army. Auxilia points cost more than regular legions and they may not go quietly when they are fired.

The next routine is the shipbuilding segment. As we have already said, make sure you have crews for all your ships. You can divide your ships between the navy and the merchant navy as you see fit. You will then be told how many ships you have available for overseas trade and you will be asked how many ships you wish to send overseas and how many war galleys are to escort them. Your merchant ships are always risking attacks by pirates so it is a good idea to send along war galleys as escorts. Any that you don't send overseas are assumed to be patrolling home waters protecting your local trading ships.

After you have made all these decisions, you will be advised if there are any wars or rebellions under way around the Empire. At the conclusion of this section, you will be told how well your forces fared and your losses in the conflict. Be warned, if you lose a battle your losses can be heavy! The computer will give you a complete list of your losses and any alteration to your trading value after each fight.

The next routine tells you about the losses that piracy has inflicted on your merchantmen. These losses can be reduced by protecting your merchant navy with a strong force of war galleys.

Following this you will receive a full report of your financial transactions for the year including your trading figures and your expenditure on the armed forces. The bottom line will be the amount of gold in the treasury.

Your people now need to be fed. The next routine in the program will tell you how much food your local trading ships brought back and if this was enough. If it wasn't you'll be forced to import more grain at greatly inflated prices.

The final routines for each turn cover your popularity and give you a points score based on the state of the Empire and your efforts to expand it. This score may fluctuate widely during the game, even going into negative numbers if you are doing a very poor job as Emperor.

```

10 REM *** ROMAN EMPIRE ***
20 GOSUB 200:REM INITIALISE
30 GOSUB 7000:REM START
40 GOSUB 500:REM ALLOCATE LABOUR
50 GOSUB 800:REM HIRE AUXILIA
60 GOSUB 1150:REM SHIPBUILDING
70 GOSUB 1370:REM TRADE ALLOCATION
80 GOSUB 1530:REM WARS
90 GOSUB 1800:REM WIN OR LOSE/LOSSES
100 GOSUB 2410:REM PIRACY
110 GOSUB 2550:REM TRADE
120 GOSUB 2700:REM FEED THE ROMANS
130 GOSUB 2870:REM POLITICAL SEGMENT
140 GOTO 3110:REM END TURN
200 PRINT "T":POKE 53280,0:POKE 53281,0:TR=50000
:TN=-55:PO=100000:LP=250
205 REM CLR SCRN
210 MS=INT(20*RND(1))+111
220 WS=INT(10*RND(1))+30
230 FS=INT(10*RND(1))+10
240 VL=100:FR=100:FL=1
250 LA=INT(10*RND(1))+6
260 MA=INT(6*RND(1))+5
270 LE=INT(10*RND(1))+11
280 MC=2000:LN=MS:LM=MS:GV=100
290 LS=INT(30*RND(1))+16
300 UL=250-LS-LE-LM-LN-LA
310 IF UL<0 THEN GOTO 210
320 X$="#####"
325 REM HOME RIGHT CRSR
330 Y$="#####"
335 REM DOWN CRSR
340 VO=54296:AD=54277:SR=54278:WF=54276:HF=54273
:LF=54272
350 RETURN

```

```

500 IF FL=0 THEN PRINT "CHECKING LABOUR
    REQUIREMENTS"
505 REM CLR SCRN
510 FOR TD=1 TO 200:NEXT:FL=0
520 IF UL<>0 THEN GOSUB 5800
530 IF UL=0 THEN PRINT "YOU HAVE NO
    LABOUR TO ALLOCATE"
535 REM CLR SCRN DOWN CRSR RIGHT CRSR LT BLUE
    REV ON
540 IF UL=0 THEN GOSUB 6010
550 IF UL=0 THEN RETURN
560 PRINT "LABOUR TO ALLOCATE =" ;UL
565 REM CLR DOWN CRSR
570 PRINT "WHERE DO YOU WISH TO EMPLOY THEM?"
575 REM DOWN CRSR
580 PRINT "(1) = LEGIONS",,,"(2) = NAVY"
590 PRINT "(3) = MERCHANT NAVY",,,"(4) = CRAFT
    GUILDS"
600 PRINT "(5) = SHIP BUILDING"
610 INPUT A
620 IF A<1 OR A>5 THEN GOTO 560
630 IF A=1 THEN GOTO 680
640 IF A=2 THEN GOTO 710
650 IF A=3 THEN GOTO 740
660 IF A=4 THEN GOTO 770
670 IF A=5 THEN GOTO 800
680 PRINT "LABOUR FOR LEGIONS"
685 REM DOWN CRSR
690 INPUT B:IF B>UL THEN GOTO 680
700 LA=LA+B:UL=UL-B:GOTO 500
710 PRINT "LABOUR FOR NAVY"
720 INPUT B:IF B>UL THEN GOTO 720
725 REM " " IS DOWN CRSR
730 LN=LN+B:UL=UL-B:GOTO 500
740 PRINT "LABOUR FOR MERCHANT NAVY"
750 INPUT B:IF B>UL THEN GOTO 750
760 LM=LM+B:UL=UL-B:GOTO 500
770 PRINT "LABOUR FOR GUILDS"
780 INPUT B:IF B>UL THEN GOTO 780
790 LE=LE+B:UL=UL-B:GOTO 500
800 PRINT "LABOUR FOR SHIPBUILDING"
810 INPUT D:IF D>UL THEN GOTO 810
820 LS=LS+B:UL=UL-B:GOTO 500

```

```

830 PRINT "J":REM CLR SCRN
840 X=1:Y=6:GOSUB 6050
850 PRINT "NO.OF AUXILIA UNITS FOR HIRE =";MA
860 PRINT "NO.OF AUXILIA UNITS HIRED   =";MH
865 REM DOWN CRSR
870 PRINT "DO YOU WISH TO....":GOSUB 5000:
    GOSUB 6030
880 PRINT "(1) = HIRE AUXILIA UNITS"
890 PRINT "(2) = FIRE AUXILIA UNITS"
900 PRINT "(3) = NONE OF THE ABOVE";
910 INPUT B:IF B<1 OR B>3 THEN GOTO 910
920 IF B=1 THEN GOTO 950
930 IF B=2 THEN GOTO 980
940 IF B=3 THEN RETURN
950 PRINT "HOW MANY UNITS WILL YOU HIRE"
955 REM DOWN CRSR
960 INPUT B:IF B>MA OR B<0 THEN GOTO 950
970 MH=MH+B:MA=MA-B:GOTO 830
980 PRINT "HOW MANY UNITS WILL YOU FIRE"
990 INPUT B:IF B>MH OR B<0 THEN GOTO 990
1000 MH=MH-B:MA=MA+B:MI=B*20:X=INT(100*RND(1)+1)
1010 PRINT"THE AUXILIA YOU HAVE FIRED ARE",
    "THREATENING TO MARCH ON ROME!!"
1015 REM CLR SCRN DOWN CRSR LT RED YELLOW
1020 GOSUB 6000:IF MICK THEN GOTO 1120
1030 PRINT "THE AUXILIA ARE RIOTING!":
    GOSUB 5200:GOSUB 6000
1035 REM DOWN CRSR REV ON RIGHT CRSR
1040 U=INT(10*RND(1)+1)*1000:V=INT(20*RND(1)+1)
    *1000:W=INT(10*RND(1)+1)+2
1050 PRINT "THEY HAVE LOOTED YOUR TREASURY OF"
    ;U;,"GOLD PIECES!":GOSUB 6010
1060 PRINT "AND PUT";V;"CITIZENS TO THE SWORD!!"
    ":GOSUB 6010
1070 PRINT "LABOUR KILLED:-":GOSUB 6010
1080 PRINT "CRAFT GUILDS =":INT(W/3)
1090 PRINT "SHIP BUILDING =":INT(W/3)
1100 TR=TR-U:PO=PO-V:LS=LS-INT(W/3):LE=LE-INT
    (W/3)
1110 GOSUB 6080:GOTO 830
1120 PRINT "THE AUXILIA UNITS HAVE DECIDED TO"

```

```

1130 PRINT "RETURN TO THEIR HOMELANDS
      PEACEFULLY"
1140 GOSUB 6000:GOTO 830
1150 PRINT "J":US=LS:REM CLR SCRN
1160 IF US=0 THEN PRINT "JNOYOU HAVE NO MORE
      SHIPS TO BUILD":GOSUB 6010:RETURN
1165 REM CLR SCRN DOWN RIGHT CRSR
1170 PRINT "JSHIPS TO BUILD =" :US
1175 REM CLR SCRN
1180 NW=LN-WS:MW=LM-MS
1190 PRINT "JNO.OF CREWS WITHOUT WAR GALLEYS:
      ":NW
1195 REM DOWN CRSR
1200 PRINT "NO.OF CREWS WITHOUT MERCHANT SHIPS:
      ":MW
1210 PRINT "JDO YOU WISH TO BUILD....."
1220 PRINT "(1) = WAR GALLEYS",,,, "(2) = MERCHANT
      SHIPS"
1230 PRINT "(3) = NONE";
1240 INPUT B
1250 IF B<0 OR B>3 THEN GOTO 1240
1260 IF B=1 THEN GOTO 1290
1270 IF B=2 THEN GOTO 1320
1280 IF B=3 THEN GOTO 1350
1290 PRINT "JHOW MANY WAR GALLEYS WILL YOU
      BUILD";
1295 REM DOWN CRSR
1300 INPUT B:IF B<0 OR B>US THEN GOTO 1300
1310 US=US-B:WS=WS+B:GOTO 1160
1320 PRINT "JHOW MANY MERCHANT SHIPS WILL YOU
      BUILD";
1330 INPUT B:IF B<0 OR B>US THEN GOTO 1330
1340 US=US-B:MS=MS+B:GOTO 1160
1350 RETURN
1370 IF LN>WS THEN WA=WS
1380 IF MS>LN THEN WA=LN
1390 IF MS>LM THEN MU=LM
1400 IF LM>MS THEN MU=MS
1410 PRINT "J":GOSUB 6020
1415 REM CLR SCRN
1420 X=1:Y=5:GOSUB 6050
1430 PRINT "WAR GALLEYS AVAILABLE =" :WA
1440 PRINT "MERCHANT SHIPS AVAILABLE =" :MU

```

```

1450 PRINT "HOW MANY MERCHANT SHIPS TO". "TRADE OVERSEAS"
1460 PRINT "HOW MANY WILL YOU SEND OVERSEAS TO TRADE"
1465 REM DOWN CRSE
1470 INPUT B IF B<0 OR B>LE THEN GOTO 1470
1480 NO=B:ML=NU-B
1490 PRINT "HOW MANY WAR GALLEYS WILL ESCORT THEM"
1500 INPUT B:IF B<0 OR B>WA THEN GOTO 1500
1510 NO=B:ML=WA-B
1520 RETURN
1530 PRINT "3":REM CLR SCRN
1540 X=5:Y=6:GOSUB 6050:PRINT "CHECKING FOR WARS AND REBELLIONS"
1545 REM REV ON PURPLE YELLOW
1550 GOSUB 6010
1560 PP=INT(1000*RND(1)+1)+(VF*3)
1570 IF PP>950 THEN X=5:Y=10:GOSUB 6050:VF=VF+5
1580 IFPP>950THENPRINT"THERE ARE NO WARS OR REBELLIONS":GOSUB 6020:GOTO 1740
1590 IF PP<175 THEN A$="WAR":B$="WITH CARTHAGE":GOTO 1690
1600 IF PP<350 THEN A$="REBELLION":B$="IN GAUL":GOTO 1700
1610 IF PP<450 THEN A$="REBELLION":B$="IN BRITAIN":GOTO 1700
1620 IF PP<525 THEN A$="WAR":B$="WITH MACEDONIA":GOTO 1710
1630 IF PP<625 THEN A$="WAR":B$="WITH PARTHIA":GOTO 1710
1640 IF PP<675 THEN A$="WAR":B$="WITH GREECE":GOTO 1720
1650 IF PP<750 THEN A$="WAR":B$="WITH PICTS":GOTO 1720
1660 IF PP<875 THEN A$="REBELLION":B$="IN JUDEA":GOTO 1720
1670 IF PP<925 THEN A$="REBELLION":B$="IN SPAIN":GOTO 1730
1680 A$="REBELLION":B$="ON CRETE":GOTO 1730
1690 C=INT(15*RND(1))+5:H=1.8:GOTO 1750

```



```

1700 G=INT(12*RND(1))+3:H=1.6:GOTO 1750
1710 G=INT(10*RND(1))+2:H=1.4:GOTO 1750
1720 G=INT(8*RND(1))+1:H=1.2:GOTO 1750
1730 G=INT(4*RND(1))+1:H=1:GOTO 1750
1740 G=INT(3*RND(1))+1:GOSUB 6030:PX=100:FX=100
:RETURN
1750 X=C:Y=10:GOSUB 6050
1760 PRINT A$:" BREAKS OUT ":B$:GOSUB 5200
1770 GOSUB 6080:RETURN
1800 IF PP>950 THEN GOTO 2280
1810 ZD=INT(100*RND(1))+1
1820 IF THK<50 THEN ZD=2:GOTO 1860
1830 IF THK<45 THEN ZD=1.5:GOTO 1860
1840 IF THK<40 THEN ZD=1:GOTO 1860
1850 IF THK<35 THEN ZD=.75
1860 ZC=INT((LA+MH+(WL/3))*ZD/H)
1870 IF ZC>100 THEN ZC=99
1880 PRINT "C"
1885 REM CLR SCRN
1890 X=8:Y=1:GOSUB 6050:PRINT "YOUR CHANCE TO
WIN IS":ZC:"%"
1910 GOSUB 5200:GOSUB 6010
1920 IF ZB<ZC THEN GOTO 2050
1930 PRINT "***** ROME IS DEFEATED
*****":GOSUB 5400
1935 REM DOWN CRSR REV ON WHITE YELLOW
1940 WW=INT(10*RND(1)+1*H):IF WW>WL THEN WW=WL
1950 MW=INT(20*RND(1)+7*H):IF MW>ML THEN MW=ML
1960 MX=INT(5*RND(1)+2*H):IF MX>MO THEN MX=MO
1970 WX=INT(3*RND(1)+1*H):IF WX>WO THEN WX=WO
1980 AW=INT(6*RND(1)+1*H):IF AW>LA THEN AW=LA
1990 SW=INT(8*RND(1)+1*H):IF SW>MH THEN SW=MH
2000 FX=INT(20*RND(1)+79)
2010 PX=INT(5*RND(1)+95)
2020 SX=INT(3*RND(1)*H):IF SX>LS THEN SX=LS
2030 EX=INT(3*RND(1)*H):IF EX>LE THEN EX=LE
2040 VF=VF-INT(G*2):GOTO 2160
2050 PRINT "***** ROME IS VICTORIOUS
*****":GOSUB 5400
2055 REM DOWN CRSR REV ON LT RED YELLOW
2060 WW=INT(5*RND(1)+1*H):IF WW>WL THEN WW=WL
2070 MW=INT(10*RND(1)+3*H):IF MW>ML THEN MW=ML
2080 MX=1:IF MO=0 THEN MX=0

```

```

2090 WX=0
2100 AW=INT(2*RND(1)+1*H):IF AW>LA THEN AW=LA
2110 SW=INT(3*RND(1)+1*H):IF SW>MH THEN SW=MH
2120 FX=100
2130 PX=100
2140 SX=0
2150 EX=0:VF=VF+INT(G*1.5)
2160 PRINT:PRINT "200LOSSES"
2165 REM REV ON RIGHT CRSR ORANGE DOWN CRSR
      YELLOW
2170 PRINT MW:"LOCAL WARSHIPS"
2180 PRINT MX:"OVERSEAS WARSHIPS"
2190 PRINT:PRINT MW:"LOCAL MERCHANT SHIPS"
2200 PRINT MX:"OVERSEAS MERCHANT SHIPS"
2210 PRINT:PRINT AW:"LEGIONS"
2220 PRINT SW:"AUXILIA UNITS"
2230 PRINT:PRINT 100-FX:"% OF FOOD STOCKPILE"
2240 PRINT 100-PX:"% OF THE EMPIRE'S POPULATION"
2250 PRINT:PRINT SX:"SHIPBUILDING COHORTS"
2260 PRINT EX:"CRAFTSMAN GUILDS"
2270 PRINT:PRINT:GOSUB 6080
2280 PRINT "3":X=3:Y=10:GOSUB 6050
2285 REM CLR SCRN
2290 IF ZB<=ZC THEN PRINT "ROME'S TRADE VALUE
      INCREASES BY";G;"%"
2300 IF ZB<=ZC THEN VL=VL+G
2310 IF ZB>ZC THEN PRINT "ROME'S TRADE VALUE
      DECREASES BY";G;"%"
2320 IF ZB>ZC THEN VL=VL-G
2330 PRINT:PRINT TAB(8) "TRADE VALUE IS NOW";
      VL;"%"
2340 GOSUB 6080
2350 WS=WS-MW-WX:LN=LN-MW-WX
2360 MS=MS-MW-MX:LM=LM-MW-MX
2370 LA=LA-AW:MH=MH-SW
2380 FS=INT(FS*FX/100)
2390 PO=INT(PO*PX/100)
2400 LS=LS-SX:LE=LE-EX:RETURN
2410 PRINT"4":X=7:Y=6:GOSUB 6050
2420 PRINT "4PIRATES ATTACK YOUR SHIPPING":
      GOSUB 5000
2430 RL=INT(50*RND(1)+50)

```

```

2435 REM REV ON WHITE YELLOW
2440 RP=INT((RL-WL)/10)
2450 IF RP<0 THEN RP=0
2460 IF RP>NL THEN RP=ML
2470 ML=ML-RP:MS=MS-RP:LM=LM-RP
2480 PRINT:PRINT RP;"LOCAL MERCHANT SHIPS LOST"
2490 RQ=INT(50*RND(1)+50)
2500 RQ=INT((RQ-MQ)/10)
2510 IF RQ<0 THEN RQ=0
2520 IF RQ>MQ THEN RQ=MQ:MQ=MQ-RQ:MS=MS-RQ:LM=
    LM-RQ
2530 PRINT:PRINT RQ;"OVERSEAS MERCHANT SHIPS
    LOST"
2540 GOSUB 6010:RETURN
2550 PRINT "J":X=3:Y=1:GOSUB 6050
2560 PRINT "FINANCIAL RETURN"
2565 REM REV ON LT RED YELLOW DOWN CRSR
2570 PRINT INT(ML*VL);"= LOCAL TRADE RETURN"
2580 PRINT INT(MQ*VL*20);"= OVERSEAS TRADE
    RETURN"
2590 TR=TR+INT((ML*VL)+(MQ*VL*20))
2600 PRINT TR;"= GOLD IN TREASURY"
2610 GOSUB 6020:X=3:Y=9:GOSUB 6050
2620 PRINT "EXPENSES":PRINT
2625 REM REV ON LT BLUE YELLOW
2630 PRINT (MH*2500);"FOR";MH;"AUXILIA UNITS"
2640 PRINT LA*500;"FOR";LA;"LEGIONS"
2650 PRINT LN*500;"FOR";LN;"NAVAL CREWS":PRINT
    ""
2660 TR=TR-(LA*500)-(LN*500)-(MH*2500)
2670 PRINT "FINAL BALANCE":PRINT
2675 REM REV ON PURPLE YELLOW
2680 PRINT TR;"GOLD PIECES"
2690 GOSUB 6080:RETURN
2700 FV=INT(40*RND(1))+85
2720 PRINT "J":X=10:Y=2:GOSUB 6050
2730 PRINT "FEEDING YOUR PEOPLE"
2735 REM REV ON LT GREEN DOWN CRSR YELLOW
2740 PRINT "YOUR";ML;"LOCAL MERCHANT SHIPS"
2750 PRINT "HAVE PURCHASED";INT(ML*FV/100);
    "FOOD POINTS"
2760 PRINT "PLUS YOUR FOOD STOCKPILE OF"
2770 PRINT FS;"POINTS GIVES YOU";FS+INT(ML*FV/

```

```

100)"FOOD POINTS"
2780 FS=FS+INT(ML*FV/100)
2790 FS=FS-FR:IF FS<0 THEN GOTO 2810
2800 PRINT "THERE IS PLENTY OF FOOD FOR ALL"
      :GOSUB 6080:RETURN
2810 PRINT "YOU HAVE STARVED THE
      PEOPLE!!"
2820 PRINT "THEY NEED";ABS(FS);"FOOD POINTS"
2830 GOSUB 6080:TR=TR+(2000*FS)
2840 PRINT "PAY ";-2000*FS;"GOLD PIECES TO
      IMPORT","GRAIN AT INFLATED PRICES"
2850 VF=VF-(2*FS):FS=0
2860 GOSUB 6080:RETURN
2870 PRINT "POPULARITY REPORT":
      X=1:Y=5:GOSUB 6050
2875 REM REV ON CLR SCRN RIGHT CRSR DOWN CRSR
      CYAN YELLOW
2890 IF VF<0 THEN GOTO 3010
2990 PRINT "YOUR RULE IS STILL POPULAR"
3000 GOSUB 6080:RETURN
3010 PRINT "IMPERIAL INTELLIGENCE REPORTS"
3020 PRINT "THAT YOUR RULE IS NOT POPULAR"
3030 PRINT"YOU NEED";ABS(VF);"POINTS TO BECOME
      POPULAR":IF TR<1000 THEN RETURN
3035 REM DOWN CRSR
3040 PRINT "HOW MUCH WILL YOU SPEND TO APPEASE"
3050 PRINT "ROME AND ITS CITIZENS";
3060 INPUT A
3070 IF A=0 THEN RETURN
3080 A=INT(A/1000)
3090 VF=VF+A
3100 RETURN
3110 PO=INT(PO*1.07):FR=INT(PO/1000):WW=0:MW=0:
      WX=0:MX=0:AW=0:SW=0:FX=0
3120 MA=MA+INT(6*RND(1)+1):PX=0:SX=0:EX=0
3130 TN=TN+1:UL=INT(PO/2500)
3140 SC=INT(((LA+LN+LM+LE+LS)*100)+(TR/2)+(PO/
      50)/(100-TN))
3145 PRINT "YOUR SCORE =";SC:GOSUB
      6080
3147 REM CLR SCRN DOWN CRSR RIGHT CRSR
3150 IF TND=-36 THEN GOTO 3170

```

```

3160 X=1:Y=10:GOSUB 6050:GOTO 40
3170 FOR TD=1 TO 1500:NEXT:PRINT "J":REM CLR
      SCRN
3175 X=12:Y=12:GOSUB 6050
3180 PRINT "THE GAME IS OVER!!!"
3185 X=8:Y=14:GOSUB 6050
3190 PRINT "YOU SCORED";SC;"POINTS"
3200 END
5000 REM SOUND
5010 POKE AD,17:POKE SR,132:POKE WF,17
5020 POKE VO,15:POKE HF,68:POKE LF,149
5030 FOR TD=1 TO 350:NEXT
5040 POKE VO,0:POKE WF,0
5050 RETURN
5200 REM SOUND
5210 POKE AD,17:POKE SR,192:POKE WF,33
5220 POKE VO,15
5230 POKE HF,5:POKE LF,155:FOR TD=1 TO 1000:NEXT
5240 POKE WF,0:POKE VO,0:RETURN
5400 REM SOUND
5410 POKE VO,15:POKE AD,18:POKE SR,64:POKE WF,17
5420 FOR TT=1 TO 3:POKE HF,102:POKE LF,194:FOR
      TD=1 TO 150:NEXT
5430 POKE HF,0:POKE LF,0:FOR TD=1 TO 150:NEXT:
      NEXT
5440 POKE VO,0:POKE WF,0:RETURN
5800 PRINT "JSENATE REPORT:YEAR";ABS(TN);
      "B.C.##### "
5805 PRINT "J" TAB(26) "J" REM
      14 SPACES
5810 PRINT "JPOPULATION",PD
5820 PRINT "ARMY LEGIONS",LA
5830 PRINT "AUXILIA UNITS",MH
5840 PRINT "JMAR GALLEYS",MS
5850 PRINT "NAVY CREWS",LN
5860 PRINT "JMERCHANT SHIPS",MS
5870 PRINT "MERCHANT SHIP CREWS",LM
5880 PRINT "JSHIPBUILDING COHORTS";LS
5890 PRINT "CRAFTSMAN GUILDS",LE
5900 PRINT "JUNALLOCATED LABOUR",UL
5910 PRINT "JGOLD IN TREASURY",TR
5920 PRINT "FOOD STOCKPILE",FS

```

```

5930 PRINT "FOOD REQUIRED",FR
5950 GOSUB 6080:RETURN
6000 FOR TD=1 TO 1000:NEXT TD
6010 FOR TD=1 TO 500:NEXT TD
6020 FOR TD=1 TO 475:NEXT TD
6030 FOR TD=1 TO 25:NEXT TD
6040 RETURN
6050 PRINT LEFT$(X$,X);LEFT$(Y$,Y);
6070 RETURN
6080 X=10:Y=22:GOSUB 6050:PRINT "HIT ANY KEY
      TO CONTINUE":GOSUB 5000
6090 GET A$
6100 IF A$="" THEN GOTO 6090
6110 RETURN
7000 POKE 53280,6:POKE 53281,7
7010 XX$="                ROMAN EMPIRE"
7020 GX=26
7030 FOR XG=1 TO 26
7040 G2$=MID$(XX$,GX,XG)
7050 PRINT "XXXXXXXXXXXX";G2$;
7060 GOSUB 7400
7100 GX=GX-1
7110 NEXT
7120 POKE V0,0:POKE WF,0:POKE HF,0:POKE LF,0
7130 FOR TD=1 TO 1000:NEXT
7140 PRINT "J":POKE 53280,3:POKE 53281,6
7150 RETURN
7400 POKE V0,15:POKE AD,17:POKE SR,18:POKE WF,17
7410 POKE HF,INT(100*RND(1)+75)
7420 POKE LF,INT(100*RND(1)+120)
7430 FOR TD=1 TO 50:NEXT
7450 RETURN

```

BATTLE OF BRITAIN

This program recreates the first fourteen days of September, 1940. You are in command of the 11th Group, Royal Air Force and your duty is to deploy your Spitfires and Hurricanes to prevent the Luftwaffe from bombing its chosen targets.

At the beginning of each day you are given an intelligence report which shows an estimate of the total number of serviceable enemy aircraft available and the state of readiness of your airfields. These factors continually change throughout the game. Aircraft will be lost in combat, your rate of replacement aircraft becoming available will alter if your aircraft factories are bombed and your serviceability will be affected if your airfields and early warning radar stations are attacked.

After you have read through your intelligence report, the Luftwaffe will choose its targets for the day's raids. The screen will clear and you will be given the opportunity to deploy your aircraft to provide the best possible defence.

It is extremely important to get as many of your aircraft as possible into action. The number of fighters which manage to intercept the incoming bombers will be reduced by poor deployment and by bad weather. You will be given a weather report each turn which will show you the amount of cloud cover. On days with heavy cloud your aircraft deployment is very important.

If the Luftwaffe carries out successful attacks on your Radar stations or on your airfields your level of serviceability will be drastically reduced on the next turn. This will affect the number of fighters you have available and the number of aircraft which are likely to intercept the Luftwaffe bombers. Raids on the aircraft production factories will reduce the number of replacement aircraft being built.

Although the odds are heavily against you, careful planning and a degree of good luck will bring you a victory.

The program is controlled by the main loop in lines 40 to 110. Building up the program in modules such as this makes for better

running of the program and also aids debugging when a problem arises.

Line 20 calls up the initialisation routine. Strings are set up to hold the names of all the aircraft and various other words which are used throughout the game. These are then used in PRINT statements. This saves a lot of repetitive typing and makes the statements more flexible. Lines 5050 to 5080 give the starting strengths of all the aircraft and also initialise the variables used to work out the serviceability of the British fighters (BS) and the German aircraft (GS). The variable DD is used to count the number of days played during the course of the game. Lines 5090 and 5100 set up the cursor strings which are used to position PRINT statements on the screen.

In lines 5200 to 5240 we have a series of FOR/NEXT loops which are used to provide time delays in the program. Loops of different lengths can be obtained by using GOSUB and calling up different line numbers. For instance GOSUB 5200 will provide a delay of 1 to 2000 while GOSUB 5220 will give you a delay of 1 to 500. Line 30 calls up the start routine and, after it is over, we then go into the main game loop.

Line 40 calls up the intelligence report display at the start of each turn. Lines 1010 to 1070 work out and display the estimate of serviceable Luftwaffe aircraft. This number is roughly based on the actual number of bombers and fighters available in France at the time. Line 1090 and 1110 perform the same task for the British fighters. Once again we have tried to keep to the number of Spitfires and Hurricanes which the 11 Group had available.

The routine from line 1500 works out the Luftwaffe targets each turn. We are not going to give too much away about how this is done, but the method creates results which are related to historical fact. Line 1580 works out the weather factor which is used to decide the cloud cover. Heavy cloud will greatly reduce the number of fighters which go into action.

Lines 2000 to 2400 allow you to choose the area you are going to defend. This routine then compares the Luftwaffe plan of attack to your choice and works out what percentage of your fighters will intercept the bombers.

Lines 2500 to 3110 contain the battle routines. Each aircraft type is given a percentage chance of shooting down an enemy aircraft. The weather factor, the degree of serviceability and the interception rate, (IR), is then used to work out how many of each type of aircraft joined in the battle. Lines 2900 to 3110 then use the percentage chance to work out victories as each type of plane goes into battle. The results will print up on the screen aircraft by aircraft.

Line 90 calls up the routine which controls the results of the bombing attacks and awards victory points to the Luftwaffe. Line 110 calls up the lines which provide the R.A.F. Bomber Command reprisal raids and award victory points to the R.A.F.

Line 3910 checks to see if the game is over, and if so, sends the program to the end routines. If you find that the game is too difficult to win, you can improve your chances by adding ten to the numbers in lines 3940 to 3990.

If the game is still in progress we then go back to the start of the main loop to go through another day in September, 1941.

```
10 REM *** BATTLE OF BRITAIN ***
20 GOSUB 5000:REM INITIALISE
30 GOSUB 6000:REM START
40 GOSUB 1000
50 GOSUB 1500
60 GOSUB 2000
70 GOSUB 2500
80 GOSUB 3200
90 GOSUB 3400
100 GOSUB 3700
110 GOTO 3800
1000 PRINT "J" TAB(7) "R.A.F. INTELLIGENCE
    REPORT"
1002 REM CLR SCRN RED BLUE DOWN CRSR
1005 PRINT TAB(14) "SEPTEMBER";DD;" "
1007 REM REV ON RED REV OFF BLUE DOWN CRSR
1010 PRINT "ESTIMATE OF SERVICEABLE ";LU$;"
    PLANES":PRINT
```

```

1020 PRINT B9$;"=";INT(B9*GS*(40*RND(1)+80)/100)
1030 PRINT B0$;"=";INT(B0*GS*(40*RND(1)+80)/100)
1040 PRINT HE$;"=";INT(H1*GS*(40*RND(1)+80)/100)
1050 PRINT D0$;"=";INT(D7*GS*(40*RND(1)+80)/
100)
1060 PRINT JU$;"=";INT(J8*GS*(40*RND(1)+80)/
100)
1070 PRINT ST$;"=";INT(J7*GS*(40*RND(1)+80)/
100):PRINT
1080 PRINT TAB(8) "11 GROUP FORCES AVAILABLE:"
:PRINT
1085 REM RED BLUE
1090 PRINT SP$;"=";INT(SP*BS/100)
1100 PRINT HU$;"=";INT(HU*BS/100)
1110 PRINT BF$;"=";BF
1120 PRINT:PRINT "R.A.F. AIRFIELD
SERVICEABILITY=";BS;"%"
1130 X=9:Y=22:GOSUB 5300
1140 GOSUB 5260:RETURN
1500 PRINT "THE ";LU$;" IS PLANNING
ITS ATTACKS:"
1505 REM BLACK CLR SCRN DOWN CRSR RIGHT CRSR
BLUE
1510 GC=INT(2000*RND(1)+1)+(LL*2)+CF
1520 IF GC<300 THEN GT=1:GC$="RADAR STATIONS":
GOTO 1580
1530 IF GC<1200 THEN GT=3:GC$="FIGHTER
AIRFIELDS":GOTO 1580
1540 IF GC<1500 THEN GT=4:GC$="FIGHTER
PRODUCTION FACTORIES":GOTO 1580
1550 IF GC<1800 THEN GT=2:GC$="ENGLISH CHANNEL
PORTS":GOTO 1580
1560 IF GC<2001 THEN GT=5:GC$="SMALL INDUSTRIAL
TOWNS":GOTO 1580
1570 GT=6:GC$="LONDON"
1580 WF=150-INT(100*RND(1)+1):IF WF>100 THEN
WF=100
1590 GOSUB 5210:RETURN
2000 PRINT "I":PRINT TAB(7) "11 GROUP
DEPLOYMENT CHOICES:"GOSUB 5220

```

```

2005 REM CLR SCRN REV ON RED BLUE DOWN CRSR
2010 PRINT "WEATHER REPORT:-";100-WF;"% CLOUD
    COVER"
2015 REM DOWN CRSR
2020 PRINT LU$;" LOSSES =";LL:PRINT
2030 PRINT "THE ";LU$;" HAS ABANDONED ATTACKS
    ON:"
2040 IF (LL*2)+CF<301 THEN PRINT "NOTHING".
2050 IF (LL*2)+CF>300 THEN PRINT "RADAR
    STATIONS"
2060 IF (LL*2)+CF>1199 THEN PRINT "11 GROUP
    FIGHTER AIRFIELDS"
2070 IF (LL*2)+CF>1799 THEN PRINT "CHANNEL
    PORTS"
2080 IF (LL*2)+CF>2000 THEN PRINT "SMALL
    INDUSTRIAL TOWNS"
2090 X=1:Y=11:GOSUB 5300
2100 PRINT "WHERE WILL YOU CONCENTRATE 11
    GROUP'S","DEFENCES?":PRINT
2110 PRINT "1=RADAR STATIONS","4=FIGHTER
    FACTORIES"
2120 PRINT "2=CHANNEL PORTS","5=SMALL TOWNS"
2130 PRINT "3=FIGHTER AIRFIELDS","6=LONDON"
2150 INPUT D
2160 IF D<1 OR D>6 THEN GOTO 2150
2170 GOSUB 5220
2180 PRINT "D":X=1:Y=3:GOSUB 5300
2185 REM CLR SCRN
2190 PRINT "YOU HAVE CHOSEN TO DEFEND"
2200 IF D=1 THEN PRINT "THE RADAR STATIONS"
2210 IF D=2 THEN PRINT "THE CHANNEL PORTS"
2220 IF D=3 THEN PRINT "THE FIGHTER AIRFIELDS"
2230 IF D=4 THEN PRINT "THE FIGHTER FACTORIES"
2240 IF D=5 THEN PRINT "THE SMALL INDUSTRIAL
    TOWNS"
2250 IF D=6 THEN PRINT "LONDON"
2260 GOSUB 5230
2270 PRINT:PRINT "THE ";LU$;" IS CONCENTRATING
    ITS"
2280 PRINT "ATTACK ON ";GC$:PRINT
2290 IF GT=D THEN IR=100

```

```

2300 IF GT-D=1 OR GT-D=-1 THEN IR=80
2310 IF GT-D=2 OR GT-D=-2 THEN IR=60
2320 IF GT-D=3 OR GT-D=-3 THEN IR=40
2330 IF GT-D>3 OR GT-D<-3 THEN IR=20
2340 PRINT IR;"% OF YOUR FIGHTERS INTERCEPT":
      PRINT:GOSUB 5220
2350 IF IR=100 THEN PRINT "11 GROUP HAS BEEN
      BRILLIANTLY DEPLOYED!"
2360 IF IR=80 THEN PRINT "11 GROUP HAS BEEN
      WELL DEPLOYED"
2370 IF IR=60 THEN PRINT "11 GROUP FACES HEAVY
      ODDS"
2380 IF IR=40 THEN PRINT "11 GROUP HAS BEEN
      POORLY DEPLOYED"
2390 IF IR=20 THEN PRINT "11 GROUP HAS BEEN
      TOTALLY OUTWITTED"
2400 GOSUB 5200:RETURN
2500 ZY=INT(SP*BS*WF/1000000*IR)+1
2510 KF=12:AZ#=SP#:BT=1
2520 GOSUB 2900
2530 ZY=INT(HU*BS*WF/1000000*IR)+1
2540 KF=7:AZ#=HU#:BT=1
2550 GOSUB 2900
2560 ZY=INT(BF*GT*WF/600)+1
2570 KF=3:AZ#=BF#:BT=1
2580 GOSUB 2900
2590 ZY=INT(B9*GS*WF/100*.65)+1
2600 KF=14:AZ#=B9#:BT=0
2610 GOSUB 2900
2620 ZY=INT(B0*GS*WF/100*.75)+1
2630 KF=5:AZ#=B0#:BT=0
2640 GOSUB 2900
2650 ZY=INT(H1*GS*WF/100*.4)+1
2660 KF=2:AZ#=HE#:BT=0
2670 GOSUB 2900
2680 ZY=INT(D7*GS*WF/100*.4)+1
2690 AZ#=D0#:BT=0
2700 GOSUB 2900
2710 ZY=INT(J7*GS*WF/100*.3)+1
2720 AZ#=ST#:BT=0
2730 GOSUB 2900
2740 ZY=INT(J8*GS*WF/100*.4)+1

```

```

2750 KF=3:AZ$=JU$:BT=0
2760 GOSUB 2900
2770 RETURN
2900 PRINT "□"
2905 REM CLR SCRN
2910 PRINT "■";ZY;AZ$;"S ATTACK■":GOSUB 5220
2915 REM RED BLUE
2920 FOR ZZ=1 TO ZY
2930 IF INT(100*RND(1))+1<KF THEN KI=1
2940 IF KI=0 THEN PRINT AZ$;ZZ;"....."
2950 IF KI=1ANDBT=1 THEN GOSUB 3000:PRINT AZ$;
    ZZ;"■";BZ$;"■":GOSUB 5500:PH=PH+1
2960 IF KI=1ANDBT=0 THEN GOSUB 3080:PRINT AZ$;
    ZZ;"■";BZ$;"■":GOSUB 5500:PH=PH+1
2970 KI=0:GOSUB 5240:NEXT ZZ
2980 PRINT "□":PRINT "■"AZ$;"S DESTROY":PH;
    "PLANES■"
2985 REM CLR SCRN RED BLUE
2990 PH=0:GOSUB 5200:RETURN
3000 SD=INT(100*RND(1))+1
3010 IF SD<20 THEN D7=D7-1:BZ$=D0$:L1=L1+1:
    GOTO 3070
3020 IF SD<44 THEN H1=H1-1:BZ$=HE$:L2=L2+1:GOTO
    3070
3030 IF SD<61 THEN J8=J8-1:BZ$=JU$:L3=L3+1:GOTO
    3070
3040 IF SD<71 THEN J7=J7-1:BZ$=ST$:L4=L4+1:GOTO
    3070
3050 IF SD<84 THEN B9=B9-1:BZ$=B9$:L5=L5+1:GOTO
    3070
3060 B0=B0-1:BZ$=B0$:L6=L6+1
3070 LL=LL+1:L0=L0+1:RETURN
3080 SD=INT(100*RND(1))+1
3090 IF SD<76 THEN HU=HU-1:BZ$=HU$:L7=L7+1:GOTO
    3110
3100 SP=SP-1:BZ$=SP$:L8=L8+1
3110 L9=L9+1:RETURN
3200 PRINT "□"
3205 REM CLR SCRN
3210 PRINT "✖✖AIR LOSSES-SEPTEMBER";DD
3215 REM REV ON RED
3220 PRINT "✖✖R.A.F.✖✖"

```

```

3225 REM DOWN CRSR REV ON DOWN CRSR BLUE
3230 PRINT SP$;"  =";L6
3240 PRINT HU$;"  =";L7
3245 PRINT "TOTAL      =";L9;" "
3250 PRINT " ";LU$;" "
3255 REM DOWN CRSR REV ON BLACK DOWN CRSR BLUE
3260 PRINT HE$;"  =";L1
3270 PRINT DO$;"  =";L2
3280 PRINT JU$;"  =";L3
3290 PRINT ST$;"  =";L4
3300 PRINT B9$;"  =";L5
3310 PRINT B0$;"  =";L6
3320 PRINT "TOTAL  =";L0;" "
3325 REM DOWN CRSR RED BLUE
3330 X=8:Y=22:GOSUB 5300:GOSUB 5260
3340 L0=0:L1=0:L2=0:L3=0:L4=0
3350 L5=0:L6=0:L7=0:L8=0:L9=0
3360 RETURN
3400 PRINT " "
3405 REM CLR SCRN
3410 PRINT "THE LUFTWAFFE ATTACK"
3412 REM DOWN CRSR
3415 PRINT " ";GC$;" "
3417 REM REV ON RED BLUE
3420 GOSUB 5220:X=1:Y=6:GOSUB 5300
3430 IF GT=1 THEN GOTO 3480
3440 IF GT=2 THEN GOTO 3530
3450 IF GT=3 THEN GOTO 3560
3460 IF GT=4 THEN GOTO 3620
3470 IF GT>4 THEN GOTO 3640
3480 ZZ=INT((H1+D7+J8+(J7*2))*WF*WF*GS/5000000)
3490 BS=BS-ZZ
3500 PRINT "R.A.F. AIRFIELD SERVICEABILITY IS
      REDUCED ";
3510 PRINT "TO";BS;"%"
3520 VP=VP+ZZ:GOTO 3660
3530 ZZ=INT((H1+D7+J8+J7)*WF*WF*GS/10000000)
3540 VP=VP+ZZ:GOTO 3660
3560 KF=101-INT(BS*1.4):IF KF<1 THEN KF=1:AZ$=
      "GERMAN BOMBER":BT=0:ZY=ZZ
3565 AZ$="GERMAN BOMBER":ZY=ZZ:BT=0
3570 GOSUB 5220:GOSUB 2900
3580 ZZ=INT(ZZ/10):BS=BS-ZZ

```

```

3590 PRINT "R.A.F. AIRFIELD SERVICEABILITY IS
      REDUCED ";
3600 PRINT "TO";BS;"%"
3610 VP=VP+ZZ:GOTO 3660
3620 ZZ=INT((H1+D7+J8+J7)*WF*WF*GS/1000000)
3625 IF INT(100*RND(1)+1)<WF THEN SR=SR-1:HR=
      HR-3:X=1
3630 IF X=1 THEN PRINT "PRODUCTION DAMAGED!":
      X=0:VP=VP+ZZ:GOTO 3660
3640 ZZ=INT((H1+D7+J8)*WF*WF*GS/1000000)
3650 VP=VP+ZZ
3660 X=1:Y=12:GOSUB 5300
3670 PRINT ZZ:"VICTORY POINTS ARE AWARDED TO
      THE ";LU$
3680 GOSUB 5200:RETURN
3700 H1=H1+GR:D7=D7+GR:J8=J8+GR
3710 J7=J7+GR:B9=B9+GR:B0=B0+GR
3720 SP=SP+SR:HU=HU+HR:DD=DD+1
3730 IF GT<>1 AND GT<>3 THEN BS=BS+2
3740 RETURN
3800 PRINT "J"
3805 REM CLR SCRN
3810 IF RND(1)>.8 THEN GOTO 3860
3820 PRINT "BOMBER COMMAND RAIDS FRENCH PORTS"
3830 PRINT "ATTEMPTING TO SINK INVASION BARGES!"
3840 GOSUB 5220:PRINT "M1 VICTORY POINT IS
      AWARDED TO THE R.A.F."
3845 REM DOWN CRSR
3850 VP=VP-1:GOTO 3910
3860 PRINT "THE PRIME MINISTER HAS ORDERED
      BOMBER"
3870 PRINT "COMMAND TO LAUNCH REPRISAL RAIDS
      AGAINST"
3880 PRINT "BERLIN!"
3890 PRINT "THE FUHRER IS FURIOUS"
3900 CF=CF+100
3910 GOSUB 5200:IF DD<15 THEN GOTO 40
3920 PRINT "J"
3925 REM CLR SCRN
3930 PRINT "THE GAME IS OVER"
3935 REM REV ON RED BLUE
3940 IF VPC46 THEN A$="R.A.F.":B$="TACTICAL"

```

```

3950 IF VP<56 THEN B$="TOTAL"
3970 IF VP>55 AND VP<65 THEN GOTO 4020
3980 IF VP>64 THEN A$=LU$:B$="TACTICAL"
3990 IF VP>74 THEN B$="TOTAL"
4000 X=1:Y=10:GOSUB 5300
4010 PRINT "THE ";A$;" HAVE WON A ";B$;"
      VICTORY":GOTO 4040
4020 PRINT "THE GAME IS A DRAM"
4030 PRINT "NEITHER SIDE HAS WON AN ADVANTAGE"
4040 PRINT "THE ";LU$;" SCORED";VP;"POINTS"
4050 STOP
5000 HE$="HE 111":DO$="DO 17"
5010 JU$="JU 88":ST$="JU 87"
5020 B9$="BF 109":B0$="BF 110"
5030 SP$="SPITFIRE":HU$="HURRICANE"
5040 BF$="A.A. GUN":LU$="LUFTWAFFE"
5050 HU=560:SP=190:BF=100:HR=12
5060 D7=300:H1=400:J8=270:SR=3
5070 J7=150:B9=330:B0=230:GR=2
5080 GS=.55:BS=75:DD=1
5090 X$="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
5095 REM HOME CURSR RIGHT CURSR
5100 Y$="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
5105 REM DOWN CURSR
5110 RETURN
5200 FOR TD=1 TO 1000:NEXT TD
5210 FOR TD=1 TO 500:NEXT TD
5220 FOR TD=1 TO 400:NEXT TD
5230 FOR TD=1 TO 90:NEXT TD
5240 FOR TD=1 TO 10:NEXT TD
5250 RETURN
5260 PRINT "HIT ANY KEY TO CONTINUE"
5265 REM REV ON RED BLUE
5270 GET A$
5280 IF A$="" THEN GOTO 5270
5290 RETURN
5300 PRINT LEFT$(X$,X):LEFT$(Y$,Y):RETURN
5500 REM SOUND ROUTINE
5510 POKE 54296,15:POKE 54277,24:POKE 54278,134
      :POKE 54276,17
5520 POKE 54273,68:POKE 54272,149
5530 FOR TD=1 TO 50:NEXT TD
5540 POKE 54296,0:POKE 54276,0

```



```
5550 RETURN
6000 PRINT "D":REM CLR SCRN
6010 POKE 53280,0:POKE 53281,14
6020 X=12:Y=10:GOSUB 5300
6030 PRINT "BATTLE OF BRITAIN"
6035 REM RED WHITE BLUE
6040 POKE 54296,15
6050 POKE 54277,132:POKE 54278,194:POKE 54276,33
6060 FOR HF=10 TO 45:FOR LF=2 TO 140 STEP 4
6070 POKE 54273,HF:POKE 54272,LF
6080 NEXT:NEXT
6090 FOR TD=1 TO 250:NEXT
6100 POKE 54296,0:POKE 54276,0
6110 POKE 53280,15:RETURN
```

SUNNYBROOK FARM

This simulation program recreates all the problems and hard work involved in running a farm, trying to make enough money to pay overheads and battling with the weather, unions and equipment breakdowns.

During each year you will have to decide if you will borrow money from the bank, replace capital equipment, buy or sell land and sheep, irrigate your crops, give in to union demands, which crops you will plant and then wait for the weather to influence the size and quality of your harvest.

At the start of the game you, the hard working farmer, are given a random age between twenty seven and thirty five. The game continues, year by year, until you reach the age of fifty, when you retire.

After the title routine, the screen will display your situation sheet. This contains all the information vital to successful farm management and it should be read carefully. You begin the game with \$3000 in the bank, a tractor, one truck, a barn and one farm labourer to help you.

Sunnybrook Farm consists of 350 acres of prime land with 100 head of sheep. You will be told how much money the bank is prepared to lend to you and how much your yearly repayment (if any) will be.

You will then have an opportunity to borrow money from the bank. If you have a loan and wish to pay it out early, borrowing a negative amount of money from the bank will pay money off your loan.

After you have finished dealing with the bank, you will be advised of any items of capital equipment which have broken down and need to be replaced. Following this you will then be given the chance to buy equipment, land or sheep for the coming year. Entering a negative number at this stage will allow you to sell land or sheep if you wish to. Sheep actually make a good cash crop. Be sure to buy sheep when the price is low and sell when the price is high! You will find that your sheep also multiply quite

rapidly, so make sure that you have enough land to support the number of sheep you wish to keep.

The next step in the game is to decide if you wish to irrigate your crops. Irrigation improves your chances of a good harvest. After this decision, you must choose which crop you are going to plant. They are listed in order of cash yield.

The weather then plays its part in the game. You must sit quietly in front of your computer and watch the rainfall readout.

The Farm Labourers' Union is the next problem to confront you. It is always trying to get a better deal for its members. The relevant information will be displayed on the screen and you must decide your course of action. It is not a good idea to start giving in to union demands too early in the game.

Following this the computer works out the amount of money you have made for the year. The computer then subtracts the labourers' wages, loan repayments and general maintenance costs before going on to the next year in the life of Sunnybrook Farm.

The game ends when you reach fifty or if you happen to go bankrupt at any stage of the game.

Lines 20 to 130 control the running of the program. They call up the various routines as they are needed. Line 20 sends the program to the initialisation routine which sets up all the starting values for the game. The variable CS in line 5050 is the starting price of each head of sheep. Lines 5060 and 5070 set up the cursor strings which are used for positioning PRINT statements on the screen. Line 5080 sets up the values for the sound generator.

The GOSUB in line 30 calls up the title routine which, through a FOR/NEXT loop runs the title down the screen. The next GOSUB, in line 40, PRINTs out the statement of your circumstances.

The banking routine runs from lines 420 to 480 and is called up by the GOSUB in line 50. Lines 510 to 550 work out the interest rate and your repayments. The borrowed money is then added to the amount of money you have in the bank (variable MB).

The next routine checks to see if you need to replace any capital equipment. The breakdowns occur in lines 2670 to 2780. These items are then replaced by lines 590 to 750. The cost of replacement is automatically deducted from your bank account.

The routine from line 760 to 1150 provides the buying and selling functions of the game. This routine is quite straightforward and a careful read through will show you how it works.

The next routine allows you to irrigate your land. Following this you are shown how much land is required to support the number of sheep you have and how much is available for crops. You then choose which crop you wish to plant. This routine makes use of an exhaustion level (EL) which affects your harvest return and makes sure that sooner or later you must leave the land fallow for a year to recover.

The rainfall is worked out and displayed by the routine in lines 1470 to 1820. This routine (called up by line 100), has a direct bearing on your yearly return.

GOSUB 1830 in line 110 is the routine in which the Farm Labourers' Union tries to win its members a pay rise. You can agree to meet its demands or risk a strike which will affect your harvest return. The choice is yours.

The amount of money you have earned for the year is worked out by the routine called up by line 120. Your actual return is worked out by the calculation in line 2190. This sum takes into account all the relevant factors involved.

After RETURNing from this routine, the program drops down to line 2350 where the year is rounded off with several things happening. First of all, the increase in your flock of sheep is worked out, the program then checks to see if you have enough land to support this many sheep and makes the necessary adjustments. The price fluctuations for your sheep are also worked out.

The next section, to line 2830, works out if your farm has suffered from any of a wide range of possible disasters during the year.

Lines 2840 to 2890 firstly increase your age by one year, (FA is farmer's age), then check to see if you have reached retirement

age. If you have, we then go down into the end routine where you are told just how well you have done as a farmer.

If you are still under fifty, the program makes sure you haven't gone bankrupt and then returns to the start of the main program for another year at Sunnybrook Farm.

We haven't given away all the secrets of this program so be careful as you are farming and most of all, enjoy playing the game.

```
10 REM *** SUNNYBROOK FARM ***
20 GOSUB 5000
30 GOSUB 5500
40 GOSUB 200
50 GOSUB 420
60 GOSUB 590
70 GOSUB 760
80 GOSUB 1160
90 GOSUB 1270
100 GOSUB 1470
110 GOSUB 1830
120 GOSUB 2130
130 GOTO 2350
200 X=1:Y=3:GOSUB 5200:POKE 53280,6:POKE 53281,2
210 PRINT "MYOUR SITUATION IS:-▲"
220 REM YELLOW CYAN
230 PRINT " BANK BALANCE  $";MB
240 PRINT "  YOUR AGE IS    ";FA
250 PRINT
260 PRINT "MYOU HAVE:-▲"
265 REM YELLOW CYAN
270 PRINT TR; "TRACTOR"
280 PRINT HD; "HEADER"
290 PRINT HV; "HARVESTER"
300 PRINT TK; "TRUCK"
310 PRINT BN; "BARN":PRINT
320 PRINT "MYOU HAVE:-▲"
325 REM YELLOW CYAN
330 PRINT AC; "ACRES OF LAND"
335 PRINT SH; "SHEEP"
340 IF IB=0 THEN PRINT " NO IRRIGATION"
350 IF IB=1 THEN PRINT " IRRIGATED LAND"
360 PRINT EL; "% LAND EXHAUSTION"
```

```

370 PRINT LR: "LABOURERS"
380 PRINT " $";PR;"= WAGES PER LABOURER"
390 PRINT " $";CL;" = COST OF LAND PER ACRE"
400 PRINT " $";BL;"X=8:Y=22:GOSUB 5200:PRINT "
    = BANK LOAN LIMIT"
410 PRINT " $";RP;"X=8:Y=23:GOSUB 5200:PRINT "
    = REPAYMENTS":RETURN
420 GOSUB 5450:GOSUB 5340
430 PRINT "WILL YOU BORROW FROM THE BANK (Y/N)
    ";:INPUT A$
440 GOSUB 5450
445 IF A$<>"Y" AND A$<>"N" THEN GOTO 430
450 IF A$="N" THEN RETURN
460 GOSUB 5240
470 PRINT "HOW MUCH WILL YOU BORROW";:INPUT AB
480 GOSUB 5450
490 IF AB>BL THEN PRINT "THE BANK WON'T LEND
    THAT MUCH!":GOSUB 5420:GOSUB 5220
495 REM HOME CRSR
500 IF AB>BL THEN GOSUB 5450:GOTO 470
510 YL=50-FA
520 IF YL>20 THEN YL=20
530 TB=TB+AB:BL=BL-AB
540 RP=INT(TB*2.2/YL)
550 MB=MB+AB
560 GOSUB 5230:GOSUB 5450:GOSUB 200
570 PRINT "YOUR REPAYMENTS = $";RP
575 REM HOME CRSR
580 GOSUB 5230:GOSUB 5450:RETURN
590 YY=YY+1:IF YY<2 THEN RETURN
595 PRINT "I"
600 IF TR=1 THEN CB=CB+1:GOTO 630
610 PRINT:PRINT "YOU MUST BUY A TRACTOR (COST
    $5000)":GOSUB 5300:GOSUB 5240
620 MB=MB-5000:TR=1
630 IF TK=1 THEN CB=CB+1:GOTO 660
640 PRINT:PRINT "YOU MUST BUY A TRUCK (COST
    $10000)":GOSUB 5300:GOSUB 5240
650 MB=MB-10000:TK=1
660 IF BN=0 THEN GOTO 680
670 IF AC/BN<500 THEN GOTO 710

```

```

680 PRINT:PRINT "YOU MUST BUILD A BARN (COST $
    3000)":GOSUB 5300:GOSUB 5240
690 MB=MB-3000:BN=BN+1:CB=CB-1
700 GOTO 670
710 IF DB=1 THEN CB=CB+1:GOTO 740
720 PRINT:PRINT "YOU MUST SINK A NEW BORE (COST
    $2000)":GOSUB 5300:GOSUB 5240
730 MB=MB-2000:CB=CB-1:DB=1
740 IF CB=3 THEN PRINT "YOU DON'T NEED TO
    REPLACE ANY CAPITAL"
745 IF CB=3 THEN PRINT "EQUIPMENT THIS YEAR":
    GOSUB 5240
750 CB=0:PRINT "J":GOSUB 200:GOSUB 5460:RETURN:
    REM CLR SCRN
760 PRINT "JYOU HAVE $";MB;"IN THE BANK":PRINT
    :PRINT:PRINT:REM CLR SCRN
770 PRINT "WHAT WILL YOU BUY ?->":PRINT
775 REM YELLOW CYAN
780 PRINT "1 = HEADER ($15000)":PRINT
790 PRINT "2 = HARVESTER ($15000)":PRINT
800 PRINT "3 = LAND ($";CL;"/ACRE)":PRINT
810 PRINT "4 = SHEEP ($";CS;"/HEAD)":PRINT
820 PRINT "5 = NO MORE PURCHASES"
825 PRINT "OK(A NEGATIVE NUMBER MEANS SELL)":
    INPUT C$
827 REM DOWN CRSR
830 IF C$<"1" OR C$>"5" THEN GOTO 760
840 IF C$="1" THEN GOSUB 5450:GOTO 890
850 IF C$="2" THEN GOSUB 5450:GOTO 950
860 IF C$="3" THEN GOSUB 5450:GOTO 1010
870 IF C$="4" THEN GOSUB 5450:GOTO 1080
880 IF C$="5" THEN GOSUB 5450:GOTO 1150
890 IF HD=1 THEN PRINT "YOU ALREADY HAVE A
    HEADER!":GOSUB 5420
895 REM HOME CRSR
900 IF MB<15000 THEN PRINT "YOU DON'T HAVE
    ENOUGH MONEY!":GOSUB 5420
910 IF HD=1 OR MB<15000 THEN GOSUB 5220:GOSUB
    5450:GOTO 760

```

```

920 MD=1:MB=MB-15000
930 PRINT "YOU HAVE PURCHASED A HEADER"
935 REM HOME CRSR
940 GOSUB 5220:GOSUB 5450:GOTO 760
950 IF HV=1 THEN PRINT "YOU ALREADY HAVE A
    HARVESTER!":GOSUB 5420
960 IF MB<15000 THEN PRINT "YOU DON'T HAVE
    ENOUGH MONEY!":GOSUB 5420
970 IF HV=1 OR MB<15000 THEN GOSUB 5220:GOSUB
    5450:GOTO 760
980 HV=1:MB=MB-15000
990 PRINT "YOU HAVE PURCHASED A HARVESTER"
1000 GOSUB 5220:GOSUB 5450:GOTO 760
1010 PRINT "HOW MANY ACRES WILL YOU PURCHASE";
1020 INPUT AS:IF AS<0 AND ABS(AS)>AC THEN GOTO
    760
1030 IF AS*CL>MB THEN GOSUB 5450:PRINT "YOU CAN
    ONLY BUY";INT(MB/CL);"ACRES"
1040 IF AS*CL>MB THEN GOSUB 5420:GOSUB 5220:
    GOSUB 5450:GOTO 760
1050 AC=AC+AS:MB=MB-(AS*CL):GOSUB 5450
1055 IF AS<0 THEN PRINT "YOU HAVE SOLD";ABS(AS)
    ;"ACRES":GOTO 1070
1060 PRINT "YOU HAVE PURCHASED";AS;"ACRES"
1070 GOSUB 5220:GOSUB 5450:GOTO 760
1080 PRINT "HOW MANY SHEEP WILL YOU PURCHASE";
1085 REM HOME CRSR
1090 INPUT SS:IF SS<0 AND ABS(SS)>SH THEN GOTO
    760
1100 IF SS*CS>MB THEN GOSUB 5450:PRINT "YOU CAN
    ONLY BUY";INT(MB/CS);"SHEEP"
1110 IF SS*CS>MB THEN GOSUB 5420:GOSUB 5220:
    GOSUB 5450:GOTO 760
1120 SH=SH+SS:MB=MB-(SS*CS):GOSUB 5450
1125 IF SS<0 THEN PRINT "YOU HAVE SOLD";ABS(SS)
    ;"SHEEP":GOTO 1140
1130 PRINT "YOU HAVE PURCHASED";SS;"SHEEP"
1140 GOSUB 5220:GOSUB 5450:GOTO 760
1150 PRINT " ":RETURN
1155 REM CLR SCRN
1160 X=7:Y=6:GOSUB 5200

```



```

1170 PRINT "YOU HAVE $";MB;"IN THE BANK"
1180 GOSUB 5340:X=4:Y=10:GOSUB 5200
1190 PRINT "WILL YOU IRRIGATE YOUR LAND (Y/N)"
1200 X=10:Y=11:GOSUB 5200
1210 PRINT "(COST $2 PER ACRE)";
1220 INPUT MI$
1230 IF MI$<>"Y" AND MI$<>"N" THEN GOTO 1160
1240 IF MI$="N" THEN IB=0:IP=1
1250 IF MI$="Y" THEN IB=1:IP=1.3:MB=MB-(AC*2)
1260 GOSUB 5240:PRINT "I":GOSUB 200:RETURN
1270 GOSUB 5460:PRINT "I"
1275 REM CLR SCRN
1280 SL=INT(SH/5):PL=AC-SL
1290 PRINT "YOU HAVE";AC;"ACRES":PRINT
1300 PRINT "YOUR";SH;"SHEEP NEED";SL;"ACRES":
    PRINT "FOR GRAZING":PRINT
1310 PRINT "YOU HAVE";PL;"ACRES AVAILABLE TO"
    PRINT "PLANT YOUR CROP":PRINT
1315 REM YELLOW CYAN
1320 PRINT "THE EXHAUSTION LEVEL OF YOUR LAND IS
    :-";EL;"%":PRINT
1330 PRINT "WILL YOU PLANT:-":PRINT
1340 PRINT "1 = BARLEY"
1350 PRINT "2 = WHEAT"
1360 PRINT "3 = PEAS"
1370 PRINT "4 = FALLOW":PRINT
1380 PRINT "WHICH CROP WILL YOU PLANT";
1390 INPUT P$
1400 IF P$<>"1" AND P$<>"4" THEN GOTO 1270
1410 IF EL>99 AND P$<>"3" AND P$<>"4" THEN PRINT
    PRINT "YOUR LAND IS EXHAUSTED"
1420 IF EL>99 AND P$<>"3" AND P$<>"4" THEN GOSUB
    5420:GOSUB 5240:GOTO 1270
1430 IF P$="1" THEN GOTO 4000
1440 IF P$="2" THEN GOTO 4040
1450 IF P$="3" THEN GOTO 4080
1460 IF P$="4" THEN GOTO 4120
1470 POKE 53281,6:POKE 53280,5:GOSUB 5340
1475 X=6:Y=5:GOSUB 5200:PRINT "I AM WORKING
    OUT THE RAINFALL":GOSUB 5220

```

```

1477 REM REV ON YELLOW
1480 WW=0:FOR XX=1 TO 40
1490 IF RND(1)>.5 THEN WW=WW+1
1500 NEXT XX
1510 FOR VV=0 TO 39
1520 POKE 1904+VV,160
1530 POKE 56176+VV,10
1540 NEXT VV
1550 FOR VV=0 TO 5
1560 POKE 1664+(40*VV),160
1570 POKE 55936+(40*VV),10
1580 NEXT VV
1590 FOR VV=1 TO WW
1600 POKE 1784+VV,102
1610 POKE 56056+VV,1
1620 GOSUB 5250
1630 NEXT VV
1640 PRINT:PRINT:PRINT:PRINT TAB(4) "THIS YEARS
      RAINFALL =" ;WW;"INCHES"
1650 GOSUB 5220:GOSUB 5460
1660 POKE 53281,2:POKE 53280,6
1670 PRINT "I"
1675 REM CLR SCRN
1680 PRINT "THIS YEARS RAINFALL =" ;WW;"INCHES"
      :PRINT:PRINT
1685 REM CYAN
1690 IF WW<10 THEN PRINT "A SAVAGE DROUGHT!!!"
      :PRINT:PRINT:GOSUB 5420
1700 IF WW<10 THEN DI=INT(RND(1)*50)+1
1710 IF WW<10 THEN PRINT "CROP LOSS =" ;100-DI;
      "%":GOSUB 5210:RETURN
1720 IF WW<19 THEN PRINT "A DROUGHT!!":PRINT:
      PRINT:GOSUB 5420
1730 IF WW<19 THEN DI=50+INT(RND(1)*50)
1740 IF WW<19 THEN PRINT "CROP LOSS =" ;100-DI;
      "%":GOSUB 5210:RETURN
1750 IF WW<23 THEN PRINT "PERFECT WEATHER":
      PRINT:PRINT:GOSUB 5300
1760 IF WW<23 THEN DI=100:GOSUB 5220:RETURN
1770 IF WW<31 THEN PRINT "DAMAGING FLOODS!!!":
      PRINT:PRINT:GOSUB 5420

```

```

1780 IF MW<31 THEN DI=50+INT(RND(1)*50)
1790 IF MW<31 THEN PRINT "CROP LOSS =";100-DI;
    "%":GOSUB 5210:RETURN
1800 PRINT "HORRENDOUS FLOODING!!!":PRINT:PRINT
1810 DI=INT(RND(1)*50)+1
1820 PRINT "CROP LOSS =";100-DI;"%":GOSUB 5210:
    RETURN
1830 PRINT "I":PRINT TAB(2) "NOW YOUR FARM HANDS
    DEMAND A PAY INCREASE"
1840 IF UM=1 THEN GOTO 1910
1850 WI=INT(20*RND(1))+1
1860 PRINT TAB(7) "NOW THEY DEMAND A";WI;"% PAY
    RISE":PRINT:PRINT
1870 SC=INT(100*RND(1))+1
1880 PRINT TAB(4) "THERE IS A";SC;"% CHANCE THAT
    THEY"
1890 PRINT TAB(4) "WILL STRIKE IF IT ISN'T
    GRANTED"
1900 IF UM=0 THEN GOTO 1990
1910 GOSUB 5220
1920 PRINT TAB(2) "BECAUSE YOU REFUSED THEIR
    CLAIM LAST"
1930 PRINT TAB(2) "YEAR THE UNIONS HAVE
    INCREASED THEIR"
1940 PRINT TAB(2) "CLAIM"
1950 WI=INT(WI*1.5)+3
1960 SC=INT(SC*2)+5
1970 PRINT TAB(2) "THE UNIONS NOW DEMAND A":
    PRINT TAB(2) WI;"% PAY RISE"
1980 PRINT:PRINT "THE CHANCE OF THEM STRIKING
    IS";SC;"%"
1990 PRINT:PRINT TAB(4) "WILL YOU PAY THE
    INCREASE? (Y/N)"
2000 GET A$:IF A$="" THEN GOTO 2000
2010 IF A$="Y" THEN PR=INT(PR*WI/100)+PR:UM=0:
    SK=1:WI=0:SC=0:GOTO 2110
2020 IF A$<>"N" THEN GOTO 1990
2030 PRINT "I":REM CLR SCREEN
2040 PRINT TAB(1) "NOW THE UNION LEADERS ARE IN
    CONSULTATION":GOSUB 5220

```

```

2045 REM DOWN CRSR
2050 UB=INT(100*RND(1))+1
2060 IF UB>SC THEN PRINT:PRINT TAB(2) "THE
      UNION HAS DECIDED NOT TO STRIKE"
2065 IF UB>SC THEN SK=1:IF UB>SC THEN GOTO 2110
2070 IF UB>SC THEN UM=1:GOTO 2110
2080 PRINT:PRINT TAB(5) "YOUR FARM HANDS ARE ON
      STRIKE!!":GOSUB 5230
2090 PRINT:PRINT TAB(1) "YOUR HARVEST RETURN
      IS REDUCED BY 30 %"
2100 UM=0:SC=0:WI=0:SK=.7
2110 GOSUB 5460
2120 RETURN
2130 PRINT "J"
2135 REM CLR SCRN
2140 IF HD=0 THEN DF=1
2150 IF HD=1 THEN DF=1.5
2160 IF HV=0 THEN VI=1
2170 IF HV=1 THEN VI=1.5
2180 PRINT TAB(2) "I AM WORKING OUT YOUR
      HARVEST RETURN":GOSUB 5220
2190 IN=INT(BP*WP*PP*FP*DI/100*DF*VI*BT*SK*PL*
      IP*20)
2200 PRINT"J"
2205 REM CLR SCRN
2210 PRINT:PRINT:PRINT TAB(6) "YOUR HARVEST
      RETURN = $":IN
2220 MB=MB+IN:GOSUB 5300:GOSUB 5220
2230 PRINT "J"
2240 PRINT:PRINT:PRINT "NEW BANK BALANCE =$":MB
2250 PRINT:PRINT "LESS WAGE COSTS OF $":(PR*LR)
2260 PRINT:PRINT "LESS BANK REPAYMENTS OF $":RP
2270 MB=MB-(PR*LR)-RP
2280 PRINT:PRINT:PRINT "LEAVES YOU WITH $":MB
2290 LR=INT(AC/500)+1
2300 X=INT(3*RND(1))+1
2310 IF X=1 THEN BL=BL+1000
2320 CL=INT(15*RND(1))*INT(CL/100)+CL
2330 GOSUB 5460
2340 RETURN
2350 X=INT(50*RND(1))+51
2360 Y=INT(X*SH/100)
2370 SH=SH+Y

```

```

2380 PRINT "J"
2385 REM CLR SCRN
2390 PRINT "YOUR FLOCK HAS INCREASED BY";Y;
    "SHEEP"
2393 REM HOME CURSR
2395 PRINT:PRINT "YOU NOW HAVE";SH;"SHEEP"
2400 IF AC*5<SH THEN SH=AC*5:PRINT
2405 IF AC*5=SH THEN PRINT "YOUR";AC;"ACRES CAN
    ONLY SUPPORT";SH;" SHEEP"
2410 GOSUB 5220:X=INT(60*RND(1))-20
2420 IF X<0 THEN A$="FALLEN"
2430 IF X>0 THEN A$="RISEN"
2440 PRINT
2450 IF X=0 THEN PRINT "THE VALUE OF YOUR SHEEP
    IS UNCHANGED":GOTO 2470
2460 PRINT "THE VALUE OF YOUR SHEEP HAS":PRINT .
    A$;" BY";ABS(X);"%
2470 GOSUB 5300:GOSUB 5240
2480 CS=INT(CS*X/100)+CS
2490 PRINT "J":X=0:Y=10:GOSUB 5200
2495 REM CLR SCRN
2500 X=INT(50*RND(1)+1)-INT(AC/40)
2510 IF X<1 THEN X=1
2520 IF X=1 THEN GOTO 2610
2530 IF X=2 THEN GOTO 2650
2540 IF X=3 THEN GOTO 2680
2550 IF X=4 THEN GOTO 2700
2560 IF X=5 THEN GOTO 2720
2570 IF X=6 THEN GOTO 2740
2580 IF X=7 THEN GOTO 2760
2590 IF X=8 THEN GOTO 2780
2600 GOTO 2830
2610 X=INT(20*RND(1)+1)
2620 Y=INT(X*SH/100)
2630 PRINT "MYSTERY DISEASE KILLS";Y;"SHEEP"
2640 SH=SH-Y:GOTO 2810
2650 X=INT(10*RND(1))+1
2660 PRINT "BLIGHT DESTROYS";X;"% OF THE CROP"
    :GOSUB 5230:PRINT"J"
2665 REM CLR SCRN
2670 BT=BT-(X/100):BB=1:GOTO 2810
2680 PRINT "ONE OF YOUR TRUCKS HAS BROKEN DOWN".

```

```

2690 TK=TK-1:GOTO 2810
2700 PRINT "A TRACTOR HAS BROKEN DOWN"
2710 TR=TR-1:GOTO 2810
2720 PRINT "YOUR BARN HAS BEEN DESTROYED"
2730 BN=BN-1:GOTO 2810
2740 PRINT "YOUR HARVESTER HAS BROKEN DOWN"
2750 HV=HV-1:GOTO 2810
2760 PRINT "YOUR HEADER HAS BROKEN DOWN"
2770 HR=HR-1:GOTO 2810
2780 PRINT "YOUR IRRIGATION BORE HAS DRIED UP"
      :GOSUB 5240
2790 DB=0
2810 GOSUB 5420
2820 GOTO 2840
2830 PRINT TAB(1) "LADY LUCK HAS SMILED ON YOU
      THIS YEAR":GOSUB 5340
2840 GOSUB 5230:FA=FA+1:BB=BB-1:IF BB<0 THEN
      BT=1:BB=0:PRINT "T"
2845 REM CLEAR SCRN
2850 IF FA=>50 THEN GOSUB 5380:X=8:Y=6:GOSUB
      5200
2855 IF FA=>50 THEN PRINT "***** YOU HAVE
      RETIRED*****"
2857 REM REV ON REV OFF REV ON REV OFF REV ON
2860 IF FA=>50 THEN GOTO 2900
2870 IF MB+BL<0 THEN GOSUB 5420:X=8:Y=6:GOSUB
      5200
2875 IF MB+BL<0 THEN PRINT "***** YOU ARE
      BANKRUPT*****"
2880 IF MB+BL<0 THEN GOTO 2950
2890 GOTO 40
2900 FS=MB+(AC*CL)+(TR*2000)+(TK*2000)+
      (BN*1000)+(HR*5000)+(HV*5000)+(SH*CS)
2910 GOSUB 5220
2920 PRINT TAB(6) "YOUR ASSETS ARE WORTH $"
      :FS
2925 REM DOWN CRSR
2930 J=INT(FS/100000)
2940 GOSUB 5230
2950 IF J=0 THEN PRINT:PRINT TAB(2) "IT'S THE

```

```

POOR HOUSE FOR YOU MY FRIEND"
2960 IF J=1 THEN PRINT:PRINT TAB(4) "YOU ARE A
    POOR STRUGGLING FARMER"
2970 IF J=1 THEN PRINT:PRINT TAB(11) "THERE
    ISN'T MUCH TO"
2980 IF J=1 THEN PRINT:PRINT TAB(9) "SHOW FOR
    ALL THOSE YEARS"
2990 IF J=2 THEN PRINT:PRINT TAB(8) "YOU ARE AN
    AVERAGE FARMER"
3000 IF J=2 THEN PRINT:PRINT TAB(10)
    "RETIREMENT WILL NOT BE"
3010 IF J=2 THEN PRINT:PRINT TAB(14)
    "UNCOMFORTABLE"
3020 IF J=3 THEN PRINT:PRINT TAB(9) "YOU ARE A
    GOOD FARMER"
3030 IF J=3 THEN PRINT:PRINT TAB(8) "YOU WILL
    LIVE IN COMFORT"
3040 IF J=3 THEN PRINT:PRINT TAB(9) "DURING
    YOUR RETIREMENT"
3050 IF J<4 THEN GOTO 3100
3060 FOR X=1 TO 6
3070 PRINT:PRINT TAB(5) "YOU ARE AN EXCELLENT
    FARMER!!"
3080 PRINT:GOSUB 5300
3090 NEXT X
3100 GOTO 3100
4000 PRINT:PRINT "YOU HAVE PLANTED BARLEY":
    GOSUB 5230
4010 BP=RND(1)*5/10+1
4020 WP=1:PP=1:FP=1
4030 EL=EL+15:PRINT "J":RETURN
4035 REM CLR SCRN
4040 PRINT:PRINT "YOU HAVE PLANTED WHEAT":GOSUB
    5230
4050 WP=RND(1)*3/10+1
4060 BP=1:PP=1:FP=1
4070 EL=EL+10:PRINT "J":RETURN
4080 PRINT:PRINT "YOU HAVE PLANTED PEAS":GOSUB
    5230

```

```

4090 FP=RND(1)*1/10+1
4100 EP=1:WP=1:FP=1
4110 PRINT "J":RETURN
4115 REM CLR SCRN
4120 PRINT PRINT "YOUR LAND IS FALLOW":GOSUB
    5230
4130 FP=0:EL=0
4140 PRINT "J":RETURN
5000 REM *** VARIABLES ***
5010 MB=3000:TR=1:BN=1:DB=1
5020 TK=1:AC=350:SH=100:BT=1
5030 CL=100:PR=2000:BL=5000
5040 FA=INT(RND(1)*8)+27
5050 LR=1:CS=INT(10*RND(1))+8
5060 X$="#####!":REM HOME RIGHT
    CURSOR*20
5070 Y$="#####":REM DOWN
    CURSOR*24
5080 VO=54296:AD=54277:SR=54278:WF=54276:HF=
    54273:LF=54272:HP=54275:LP=54274
5090 PRINT "J":RETURN:REM CLR SCRN
5100 PRINT LEFT$(X$,X):LEFT$(Y$,Y):RETURN
5110 FOR TD=1 TO 1000:NEXT TD
5120 FOR TD=1 TO 1000:NEXT TD
5130 FOR TD=1 TO 500:NEXT TD
5140 FOR TD=1 TO 400:NEXT TD
5150 FOR TD=1 TO 100:NEXT TD
5160 RETURN
5180 POKE VO,15:POKE AD,10:POKE SR,132:POKE
    WF,33
5310 POKE HF,INT(20*RND(1))+5:POKE LF,INT(50*RND
    (1))+10:FOR TD=1 TO 50:NEXT
5320 TT=TT+1:IF TT>2 THEN TT=0:POKE WF,0:POKE
    AD,0:POKE SR,0:RETURN
5325 POKE HF,0:POKE LF,0
5330 FOR TD=1 TO 25:NEXT:GOTO 5310
5340 POKE VO,15:POKE AD,10:POKE SR,144:POKE WF,
    65:POKE HP,0:POKE LP,250
5350 POKE HF,0:POKE LF,23:FOR TD=1 TO 300:NEXT
    TD
5360 POKE VO,0:POKE WF,0:POKE AD,0:POKE SR,0
5370 RETURN

```



```

5380 POKE VO,15:POKE AD,18:POKE SR,34:POKE WF,17
5390 FOR H1=10 TO 100 STEP 10:FOR L1=10 TO 100
    STEP 10
5400 POKE HF,H1:POKE LF,L1
5410 NEXT:NEXT:POKE WF,0:POKE VO,0:POKE AD,0:
    POKE SR,0:RETURN
5420 POKE VO,15:POKE AD,18:POKE SR,194:POKE
    WF,33
5430 POKE HF,4:POKE LF,73:GOSUB 5240
5440 POKE VO,0:POKE WF,0:POKE SR,0:POKE AD,0:
    RETURN
5450 PRINT "0";:FOR KK=1 TO 40:PRINT " ":NEXT
    KK:RETURN
5455 REM HOME
5460 X=0:Y=0:GOSUB 5200:PRINT "0HIT ANY KEY TO
    CONTINUE"
5470 GET B#
5480 IF B#="" THEN GOTO 5470
5490 RETURN
5500 PRINT "0" POKE 53280,13:POKE 53281,4
5505 REM CLR SCRN
5510 POKE VO,15:POKE AD,17:POKE SR,32:POKE WF,17
5515 FOR TX=1 TO 24
5520 PRINT TAB(TX) "0SUNNYBROOK FARM"
5525 REM DOWN CRSR REV ON YELLOW
5530 H1=INT(20*RND(1))+160:L1=INT(20*RND(1))+140
5540 POKE HF,H1:POKE LF,L1:FOR TD=1 TO 15:NEXT
5550 POKE HF,0:POKE LF,0
5560 NEXT:PRINT "0":REM CLR SCRN
5570 RETURN

```


CHALLENGE THE COMPUTER

3-D Noughts and Crosses

Statesman

Unscramble Alien

Reversi

Chessboard Nim



Chessboard Nim

3-D NOUGHTS AND CROSSES

This program creates a mind-stretching game of Noughts and Crosses in three dimensions. As you are playing this game you must try to visualise the board in three dimensions. If you don't, you are likely to make mistakes.

When the program is RUN you will see a title screen displayed, accompanied by a quick ascending sound routine. The screen will clear when this is finished and the playing board will be displayed. This consists of the playing area to the right of the screen and the key to the game board on the left of the screen. You will need to visualise these boards as if they are stacked on top of each other, top, middle and bottom. You are viewing them from above. The computer will use crosses as its playing pieces and you will play the noughts. In this game we have used the letter "O" instead of a zero for the noughts.

Lines 30 to 35 decide, on a random basis, who is going to have the first turn for each game played. When it is your turn the computer will prompt you. To place a piece on the board, enter the number of the board location you wish to occupy. This number is shown in the board key on the left hand side of the screen.

To win the game you must get three of your playing pieces in a row, horizontally, vertically or diagonally. This can be on one board level, as in ordinary noughts and crosses, or you can win by getting three in a row through the three boards. This can be vertically through the board, such as locations 1, 10 and 19 or 5, 14 and 23. Three locations through any of the four diagonal axes of the board will also give you a win. For example; 1, 14 and 27, or 9, 14 and 19. Please note, a combination of 1, 11 and 21 isn't a winning move. All diagonal lines through the cube must occupy location 14 on the middle board to win.

The program will play an infinite number of games and keep track of how many games each player has won. When you wish to stop playing, press the BREAK key to escape from the program.

The lines 190 to 340 print up the board and the key. The actual playing area is displayed by printing the character (using CHR\$) represented by the ASCII code which is held in each location of the A array. This value can be the ASCII code of "X", "O" or a full stop, which is used to denote an empty square.

The computer uses the long routine from lines 1030 to 2430 to make its move. This certainly isn't the most efficient way to program this routine. However it was written this way for several reasons. This form is very easy to read through and therefore easy to debug if a problem occurs. It also keeps the length of the lines under the 80 character maximum the Commodore 64 allows us to use. This avoids any messy bits of code if a line gets too long and has to be broken down to fit. Finally it is easy to type in this form. For instance, if you type line 1050 and line 1110, you can then use the screen editing functions to alter line numbers and number values to produce nearly every other line in the routine without a lot of repetitive typing.

```

10 REM *** 3-D NOUGHTS AND CROSSES ***
15 GOSUB 2970:REM *** START ROUTINE ***
20 GOTO 2820:REM INITIALISE
30 GT=RND(1):IF GT>.5 THEN PRINT "~~~~~YOU
   CAN HAVE THE FIRST MOVE"
31 REM DOWN RIGHT CRSR BLACK
33 IF GT>.5 THEN GOSUB 3200
35 IF GT>.5 THEN FOR TD=1 TO 2000:NEXT TD:GOTO
   130
40 GOSUB 190:REM PRINT BOARD
50 GOSUB 360:REM WIN CHECK
60 IF Q<>0 OR M=99 THEN 2680:REM END
70 M=0
80 GOSUB 1030:REM MAKE MOVE
90 IF M=0 OR A(M)<>E THEN M=0:GOSUB 2500
100 PRINT "I'LL MOVE IN":M
110 FOR TD=1 TO 2000:NEXT TD
120 A(M)=B
130 GOSUB 190:REM PRINT BOARD
140 GOSUB 360:REM WIN CHECK
150 IF Q<>0 OR M=99 THEN 2680:REM END
160 GOSUB 960:REM HUMAN MOVE

```

```

170 GOTO 40
180 REM *****
190 REM PRINT BOARD
200 PRINT "□":REM CLR SCRN
205 GOSUB 3300
210 PRINT TAB(8) "TOP":PRINT
215 REM RED BLUE
220 PRINT TAB(8) " 1 2 3",CHR$(A(1));" ";
    CHR$(A(2));" ";CHR$(A(3))
230 PRINT TAB(8) " 4 5 6",CHR$(A(4));" ";CHR$(
    A(5));" ";CHR$(A(6))
240 PRINT TAB(8) " 7 8 9",CHR$(A(7));" ";CHR$(
    A(8));" ";CHR$(A(9))
250 PRINT:PRINT TAB(8) "MIDDLE":PRINT
255 REM RED BLUE
260 PRINT TAB(8) "10 11 12",CHR$(A(10));" ";
    CHR$(A(11));" ";CHR$(A(12))
270 PRINT TAB(8) "13 14 15",CHR$(A(13));" ";
    CHR$(A(14));" ";CHR$(A(15))
280 PRINT TAB(8) "16 17 18",CHR$(A(16));" ";
    CHR$(A(17));" ";CHR$(A(18))
290 PRINT:PRINT TAB(8) "BOTTOM":PRINT
295 REM RED BLUE
300 PRINT TAB(8) "19 20 21",CHR$(A(19));" ";
    CHR$(A(20));" ";CHR$(A(21))
310 PRINT TAB(8) "22 23 24",CHR$(A(22));" ";
    CHR$(A(23));" ";CHR$(A(24))
320 PRINT TAB(8) "25 26 27",CHR$(A(25));" ";
    CHR$(A(26));" ";CHR$(A(27))
330 PRINT
340 RETURN
350 REM *****
360 REM WIN CHECK
370 Q=0
380 IF A(1)=E THEN 440
390 IF A(1)=A(2) AND A(2)=A(3) THEN Q=1:RETURN
400 IF A(1)=A(4) AND A(4)=A(7) THEN Q=1:RETURN
410 IF A(1)=A(5) AND A(5)=A(9) THEN Q=1:RETURN
420 IF A(1)=A(10) AND A(10)=A(19) THEN Q=1:
    RETURN

```

```

430 IF A(1)=A(14) AND A(14)=A(27) THEN Q=1:
    RETURN
440 IF A(10)=E THEN 480
450 IF A(10)=A(11) AND A(11)=A(12) THEN Q=10:
    RETURN
460 IF A(10)=A(13) AND A(13)=A(16) THEN Q=10:
    RETURN
470 IF A(10)=A(14) AND A(14)=A(18) THEN Q=10:
    RETURN
480 IF A(19)=E THEN 520
490 IF A(19)=A(20) AND A(20)=A(21) THEN Q=19:
    RETURN
500 IF A(19)=A(22) AND A(22)=A(25) THEN Q=19:
    RETURN
510 IF A(19)=A(23) AND A(23)=A(27) THEN Q=19:
    RETURN
520 IF A(2)=E THEN 550
530 IF A(2)=A(5) AND A(5)=A(8) THEN Q=2:RETURN
540 IF A(2)=A(11) AND A(11)=A(20) THEN Q=2:
    RETURN
550 IF A(11)=E THEN 570
560 IF A(11)=A(14) AND A(14)=A(17) THEN Q=11:
    RETURN
570 IF A(20)=E THEN 590
580 IF A(20)=A(23) AND A(23)=A(26) THEN Q=20:
    RETURN
590 IF A(3)=E THEN 640
600 IF A(3)=A(6) AND A(6)=A(9) THEN Q=3:RETURN
610 IF A(3)=A(12) AND A(12)=A(21) THEN Q=3:
    RETURN
620 IF A(3)=A(5) AND A(5)=A(7) THEN Q=3:RETURN
630 IF A(3)=A(14) AND A(14)=A(25) THEN Q=3:
    RETURN
640 IF A(12)=E THEN 670
650 IF A(12)=A(15) AND A(15)=A(18) THEN Q=12:
    RETURN
660 IF A(12)=A(14) AND A(14)=A(16) THEN Q=12:
    RETURN
670 IF A(21)=E THEN 700
680 IF A(21)=A(24) AND A(24)=A(27) THEN Q=21:
    RETURN

```



```

690 IF A(21)=A(23) AND A(23)=A(25) THEN Q=21:
  RETURN
700 IF A(4)=E THEN 730
710 IF A(4)=A(5) AND A(5)=A(6) THEN Q=4:RETURN
720 IF A(4)=A(13) AND A(13)=A(22) THEN Q=4:
  RETURN
730 IF A(13)=E THEN 750
740 IF A(13)=A(14) AND A(14)=A(15) THEN Q=13:
  RETURN
750 IF A(22)=E THEN 770
760 IF A(22)=A(23) AND A(23)=A(24) THEN Q=22:
  RETURN
770 IF A(6)=E THEN 790
780 IF A(6)=A(15) AND A(15)=A(24) THEN Q=6:
  RETURN
790 IF A(5)=E THEN 810
800 IF A(5)=A(14) AND A(14)=A(23) THEN Q=5:
  RETURN
810 IF A(7)=E THEN 850
820 IF A(7)=A(8) AND A(8)=A(9) THEN Q=7:RETURN
830 IF A(7)=A(16) AND A(16)=A(25) THEN Q=7:
  RETURN
840 IF A(7)=A(14) AND A(14)=A(21) THEN Q=7:
  RETURN
850 IF A(17)=E THEN 870
860 IF A(16)=A(17) AND A(17)=A(18) THEN Q=16:
  RETURN
870 IF A(25)=E THEN 890
880 IF A(25)=A(26) AND A(26)=A(27) THEN Q=25:
  RETURN
890 IF A(8)=E THEN 910
900 IF A(8)=A(17) AND A(17)=A(26) THEN Q=8:
  RETURN
910 IF A(9)=E THEN 940
920 IF A(9)=A(18) AND A(18)=A(27) THEN Q=9:
  RETURN
930 IF A(9)=A(14) AND A(14)=A(19) THEN Q=9:
  RETURN
940 RETURN
950 REM *****
960 REM ACCEPT PLAYER MOVE
970 PRINT "YOUR MOVE: "; INPUT Y

```

```

980 IF Y<1 OR Y>27 THEN 970
990 IF A(Y)◊E THEN 970
1000 A(Y)=C
1010 RETURN
1020 REM *****
1030 REM MAKE MOVE
1040 M=1:IF A(1)◊E THEN 1110
1050 IF A(1)=A(3) AND A(1)◊E THEN M=2:RETURN
1060 IF A(2)=A(3) AND A(2)◊E THEN RETURN
1070 IF A(4)=A(7) AND A(4)◊E THEN RETURN
1080 IF A(5)=A(9) AND A(5)◊E THEN RETURN
1090 IF A(10)=A(19) AND A(10)◊E THEN RETURN
1100 IF A(14)=A(27) AND A(14)◊E THEN RETURN
1110 M=3:IF A(3)◊E THEN 1170
1120 IF A(2)=A(1) AND A(2)◊E THEN RETURN
1130 IF A(6)=A(9) AND A(6)◊E THEN RETURN
1140 IF A(5)=A(7) AND A(5)◊E THEN RETURN
1150 IF A(12)=A(21) AND A(12)◊E THEN RETURN
1160 IF A(14)=A(25) AND A(14)◊E THEN RETURN
1170 M=4:IF A(4)◊E THEN 1210
1180 IF A(1)=A(7) AND A(1)◊E THEN RETURN
1190 IF A(5)=A(6) AND A(5)◊E THEN RETURN
1200 IF A(13)=A(22) AND A(13)◊E THEN RETURN
1210 M=5:IF A(5)◊E THEN 1270
1220 IF A(1)=A(9) AND A(1)◊E THEN RETURN
1230 IF A(2)=A(8) AND A(2)◊E THEN RETURN
1240 IF A(3)=A(7) AND A(3)◊E THEN RETURN
1250 IF A(4)=A(6) AND A(4)◊E THEN RETURN
1260 IF A(14)=A(23) AND A(14)◊E THEN RETURN
1270 M=6:IF A(6)◊E THEN 1310
1280 IF A(4)=A(5) AND A(4)◊E THEN RETURN
1290 IF A(3)=A(9) AND A(3)◊E THEN RETURN
1300 IF A(15)=A(24) AND A(15)◊E THEN RETURN
1310 M=7:IF A(7)◊E THEN 1360
1320 IF A(1)=A(4) AND A(1)◊E THEN RETURN
1330 IF A(8)=A(9) AND A(8)◊E THEN RETURN
1340 IF A(5)=A(3) AND A(5)◊E THEN RETURN
1350 IF A(16)=A(25) AND A(25)◊E THEN RETURN
1360 M=8:IF A(8)◊E THEN 1400
1370 IF A(7)=A(9) AND A(7)◊E THEN RETURN
1380 IF A(5)=A(2) AND A(5)◊E THEN RETURN
1390 IF A(17)=A(26) AND A(17)◊E THEN RETURN
1400 M=9:IF A(9)◊E THEN 1460

```

```

1410 IF A(7)=A(8) AND A(7)OE THEN RETURN
1420 IF A(6)=A(3) AND A(6)OE THEN RETURN
1430 IF A(1)=A(5) AND A(1)OE THEN RETURN
1440 IF A(18)=A(27) AND A(18)OE THEN RETURN
1450 IF A(14)=A(19) AND A(14)OE THEN RETURN
1460 M=10:IF A(10)OE THEN 1510
1470 IF A(11)=A(12) AND A(11)OE THEN RETURN
1480 IF A(13)=A(16) AND A(13)OE THEN RETURN
1490 IF A(14)=A(18) AND A(14)OE THEN RETURN
1500 IF A(1)=A(19) AND A(1)OE THEN RETURN
1510 M=11:IF A(11)OE THEN 1550
1520 IF A(10)=A(12) AND A(10)OE THEN RETURN
1530 IF A(14)=A(17) AND A(14)OE THEN RETURN
1540 IF A(2)=A(20) AND A(2)OE THEN RETURN
1550 M=12:IF A(12)OE THEN 1600
1560 IF A(10)=A(11) AND A(10)OE THEN RETURN
1570 IF A(15)=A(18) AND A(15)OE THEN RETURN
1580 IF A(14)=A(16) AND A(14)OE THEN RETURN
1590 IF A(3)=A(21) AND A(3)OE THEN RETURN
1600 M=20:IF A(20)OE THEN 1640
1610 IF A(19)=A(21) AND A(19)OE THEN RETURN
1620 IF A(23)=A(26) AND A(23)OE THEN RETURN
1630 IF A(2)=A(11) AND A(2)OE THEN RETURN
1640 M=21:IF A(21)OE THEN 1700
1650 IF A(19)=A(20) AND A(19)OE THEN RETURN
1660 IF A(24)=A(27) AND A(24)OE THEN RETURN
1670 IF A(23)=A(25) AND A(23)OE THEN RETURN
1680 IF A(3)=A(12) AND A(3)OE THEN RETURN
1690 IF A(14)=A(7) AND A(14)OE THEN RETURN
1700 M=16:IF A(16)OE THEN 1750
1710 IF A(10)=A(13) AND A(10)OE THEN RETURN
1720 IF A(17)=A(18) AND A(17)OE THEN RETURN
1730 IF A(14)=A(12) AND A(14)OE THEN RETURN
1740 IF A(7)=A(25) AND A(7)OE THEN RETURN
1750 M=17:IF A(17)OE THEN 1790
1760 IF A(14)=A(11) AND A(14)OE THEN RETURN
1770 IF A(16)=A(18) AND A(16)OE THEN RETURN
1780 IF A(8)=A(26) AND A(8)OE THEN RETURN
1790 M=18:IF A(18)OE THEN 1840
1800 IF A(16)=A(17) AND A(16)OE THEN RETURN
1810 IF A(15)=A(12) AND A(15)OE THEN RETURN
1820 IF A(14)=A(10) AND A(14)OE THEN RETURN
1830 IF A(9)=A(27) AND A(9)OE THEN RETURN

```

```

1840 M=19:IF A(19)◊E THEN 1900
1850 IF A(20)=A(21) AND A(20)◊E THEN RETURN
1860 IF A(22)=A(25) AND A(22)◊E THEN RETURN
1870 IF A(23)=A(27) AND A(23)◊E THEN RETURN
1880 IF A(1)=A(10) AND A(1)◊E THEN RETURN
1890 IF A(14)=A(9) AND A(14)◊E THEN RETURN
1900 M=13:IF A(13)◊E OR RND(1)>.2 THEN 1960
1910 IF A(14)=E AND A(15)=E OR A(16)=E AND A(10)
    =E THEN 1960
1920 IF A(10)=A(16) AND A(10)◊E THEN RETURN
1930 IF A(2)=A(22) AND A(2)◊E THEN RETURN
1940 IF A(14)=A(15) AND A(13)◊E THEN RETURN
1950 IF A(M)=E THEN RETURN
1960 M=14:IF A(14)◊E THEN 2050
1970 IF A(10)=A(18) AND A(10)◊E THEN RETURN
1980 IF A(12)=A(16) AND A(12)◊E THEN RETURN
1990 IF A(13)=A(15) AND A(13)◊E THEN RETURN
2000 IF A(11)=A(17) AND A(11)◊E THEN RETURN
2010 IF A(5)=A(23) AND A(5)◊E THEN RETURN
2020 IF A(1)=A(27) AND A(1)◊E THEN RETURN
2030 IF A(3)=A(25) AND A(3)◊E THEN RETURN
2040 IF A(7)=A(21) AND A(7)◊E THEN RETURN
2050 IF A(9)=A(19) AND A(9)◊E THEN RETURN
2060 M=15:IF A(15)◊E THEN 2100
2070 IF A(12)=A(18) AND A(12)◊E THEN RETURN
2080 IF A(13)=A(14) AND A(13)◊E THEN RETURN
2090 IF A(6)=A(24) AND A(6)◊E THEN RETURN
2100 M=22:IF A(22)◊E THEN 2140
2110 IF A(19)=A(25) AND A(19)◊E THEN RETURN
2120 IF A(23)=A(24) AND A(23)◊E THEN RETURN
2130 IF A(4)=A(13) AND A(4)◊E THEN RETURN
2140 M=23:IF A(23)◊E THEN 2200
2150 IF A(22)=A(24) AND A(22)◊E THEN RETURN
2160 IF A(20)=A(26) AND A(20)◊E THEN RETURN
2170 IF A(19)=A(27) AND A(19)◊E THEN RETURN
2180 IF A(15)=A(21) AND A(25)◊E THEN RETURN
2190 IF A(5)=A(14) AND A(5)◊E THEN RETURN
2200 M=24:IF A(24)◊E THEN 2240
2210 IF A(21)=A(27) AND A(21)◊E THEN RETURN
2220 IF A(22)=A(23) AND A(22)◊E THEN RETURN
2230 IF A(6)=A(15) AND A(6)◊E THEN RETURN
2240 M=25:IF A(25)◊E THEN 2290
2250 IF A(19)=A(22) AND A(19)◊E THEN RETURN

```

```

2280 IF A(20)=A(27) AND A(25) <> E THEN RETURN
2290 IF A(22)=A(21) AND A(23) <> E THEN RETURN
2295 IF A(7)=A(16) AND A(7) <> E THEN RETURN
2299 M=25 IF A(26) <> E THEN 2330
2300 IF A(23)=A(29) AND A(23) <> E THEN RETURN
2310 IF A(25)=A(27) AND A(25) <> E THEN RETURN
2320 IF A(9)=A(17) AND A(9) <> E THEN RETURN
2330 M=27 IF A(27) <> E THEN 2390
2340 IF A(25)=A(26) AND A(25) <> E THEN RETURN
2350 IF A(24)=A(21) AND A(24) <> E THEN RETURN
2360 IF A(19)=A(23) AND A(19) <> E THEN RETURN
2370 IF A(9)=A(18) AND A(9) <> E THEN RETURN
2380 IF A(1)=A(14) AND A(1) <> E THEN RETURN
2390 M=0 IF A(2) <> E THEN 2450
2400 IF A(1)=E AND A(3)=E OR A(5)=E AND A(6)=E
    THEN 2450
2410 IF A(1)=A(3) AND A(1) <> E THEN RETURN
2420 IF A(11)=A(20) AND A(11) <> E THEN RETURN
2430 IF A(5)=A(8) AND A(5) <> E THEN RETURN
2440 REM *****
2450 IF RND(1)>.7 THEN 2470
2460 IF A(14)=E THEN M=14:RETURN
2470 IF A(5)=E THEN M=5:RETURN
2480 IF A(23)=E THEN M=23:RETURN
2490 M=0
2500 Z=0
2510 Z=Z+1
2520 M=D(INT(RND(1)*12)+1)
2530 IF A(M)=E THEN RETURN
2550 IF Z<100 THEN 2510
2560 Z=0
2570 M=0
2580 Z=Z+1
2590 M=INT(RND(1)*27)+1
2600 IF A(M)=E THEN RETURN
2610 IF Z<500 THEN 2590
2620 PRINT:PRINT "IT'S A DRAW!!!"
2630 FOR TD=1 TO 2000:NEXT TD
2640 M=99
2650 RETURN
2660 GOTO 2720
2670 REM *****
2680 REM END OF GAME

```

154

```

    REM SHIFT CLR HOME 2 DOWN CRSR
3080 VO=54296:AD=54277:SR=54278:WF=54276:
    HF=54273:LF=54272
3085 HP=54275:LP=54274
3090 POKE VO,15:POKE AD,15:POKE SR,132:POKE HP,
    15:POKE LP,200:POKE WF,65
3100 H1=10:L1=10
3110 POKE HF,H1:POKE LF,L1:FOR TD=1 TO 20:NEXT
    TD
3120 H1=H1+5:L1=L1+10
3130 IF H1<100 THEN GOTO 3110
3140 POKE VO,0:POKE WF,0:H1=0:L1=0
3150 FOR TD=1 TO 1000:NEXT TD
3160 RETURN
3200 POKE VO,15:POKE AD,17:POKE SR,64:POKE WF,
    65
3210 POKE HP,10:POKE LP,1
3220 POKE HF,7:POKE LF,33
3230 FOR TD=1 TO 350:NEXT TD
3240 POKE VO,0:POKE WF,0
3250 RETURN
3300 POKE VO,15:POKE AD,18:POKE SR,132:POKE
    WF,65:POKE HP,15:POKE LP,200
3310 POKE HF,25:POKE LF,177:FOR TD=1 TO 750:
    NEXT TD
3320 POKE VO,0:POKE WF,0:RETURN

```

STATESMAN

Statesman is a boardgame distantly related to checkers. It is played on a nine by eight board and pieces are captured, not by jumping over them as in checkers, but by landing on them. The pieces move diagonally as in checkers but can move both forward and back on the board. Thus it is possible to retreat when under attack.

The computer has the first move. It will then ask "MOVE FROM". To move your piece enter the coordinates of the piece beginning with the number down the side of the board, followed by a comma and then the number across the board. For instance if you wished to move the piece at the far right of your front row you would enter 6,8. After you have done this the computer will ask "MOVE TO". Simply enter the location of the square you wish to move to in exactly the same way.

Statesman uses DATA statements to set up the board and playing pieces. This information is transferred to the screen by the routine beginning at line 500. Lines 140 to 170 set up an array and then READ the DATA statements into the elements of the array.

The routine between lines 330 and 470 work out the computer's move checking the squares around each piece for a possible capture and then a safe move if a capture isn't possible.

```
10 REM STATESMAN
15 REM "♙" IS ON Q KEY "♘" IS ON A KEY USE SHIFT
17 REM "♚" IS ON + KEY USE COMMODORE KEY
20 PRINT "Q":REM CLEAR SCREEN
25 POKE 53280,4:POKE 53281,6
30 DATA "♙123456789♙"
40 DATA "1♙♙♙♙♙♙♙1"
50 DATA "2♙♙♙♙♙♙♙2"
60 DATA "3♙♙♙♙♙♙♙3"
70 DATA "4♙♙♙♙♙♙♙4"
80 DATA "5♙♙♙♙♙♙♙5"
90 DATA "6♙♙♙♙♙♙♙6"
```



```

100 DATA "7♣♠♦♥♣♠7"
110 DATA "8♠♣♦♠♣♠8"
120 DATA "12345678910"
130 HS=0:CS=0
140 DIM S$(12,13)
150 FOR A=1 TO 10
160 READ B$
170 FOR B=1 TO 11:S$(A,B)=MID$(B$,B,1):NEXT B:
NEXT A
180 IF INT(RND(1)+.5)=0 THEN 200
190 S$(5,5)="♣":S$(5,7)=" "
200 PRINT "3":REM CLEAR SCREEN
210 GOSUB 500
220 PRINT:PRINT:PRINT
230 IF CS=6 THEN PRINT TAB(12)"I WIN I WIN!!"
:END
240 PRINT TAB(4)"MOVE FROM";
245 INPUT A,B
250 PRINT TAB(4)"MOVE TO";
255 INPUT C,D
260 IF ABS(A-C)=1 AND ABS(B-D)=1 THEN GOTO 280
270 PRINT "ILLEGAL MOVE":GOTO 240
280 IF S$(C+1,D+1)="♣" THEN HS=HS+1
290 S$(A+1,B+1)=" ":S$(C+1,D+1)="♣"
300 GOSUB 500
305 FOR X=1784 TO 1943
310 POKE X,32
315 NEXT X
320 IF HS=6 THEN PRINT TAB(14)"YOU WIN":END
330 A$="♣":GOSUB 400
340 IF FL=1 THEN GOTO 360
350 A$=" ":GOSUB 400
360 S$(E,F)=" "
370 IF S$(E+G,F+H)="♣" THEN CS=CS+1
380 S$(E+G,F+H)="♣"
390 GOTO 210
400 E=2:F=2:G=0:H=0
410 FL=0
420 IF S$(E,F) <> "♣" THEN GOTO 455
430 IF S$(E+1,F+1)=A$ THEN G=1:H=1
435 IF S$(E+1,F-1)=A$ THEN G=1:H=-1
440 IF S$(E-1,F+1)=A$ THEN G=-1:H=1
445 IF S$(E-1,F-1)=A$ THEN G=-1:H=-1

```

```

450 IF G<>0 AND H<>0 THEN FL=1:RETURN
455 E=E+1:IF E>10 THEN E=2:F=F+1
460 IF F>11 THEN RETURN
470 GOTO 410
500 PRINT "␣":FOR A=1 TO 10:PRINT:FOR B=1 TO 11
    :REM HOME CURSOR
510 PRINT "␣";:IF S$(A,B)="⬤" THEN PRINT "␣";:
    REM GREEN,LT BLUE
520 IF S$(A,B)="⬢" THEN PRINT "␣";:REM BLACK
530 IF S$(A,B)="⬢" THEN PRINT "␣";:REM LT RED
540 PRINT TAB(12) S$(A,B);
550 NEXT:NEXT
560 PRINT:PRINT TAB(4)"0000COMPUTER: ";CS,"HUMAN:
    "HS:REM DOWN CURSOR
580 RETURN

```

UNSCRAMBLE ALIEN

Unscramble Alien is a word game written by David Vendy. The object of the game is to unscramble the word bombs the alien space ship is dropping on your city. The aliens will drop ten scrambled words. You must type in the correct word to stop a bomb from hitting the city.

Use the "A" key to move your fire base to the left and the "D" key to move it to the right. Press the "S" key to fire at the alien ship. If you miss three word bombs, the game is over. This game uses sprites, sound and keyboard graphic characters to provide a very clear display.

After the title display, the instruction screens will be shown (lines 120 to 312). After you have read this information, the screen clears and lines 50000 to 50110 print your city at the bottom of the screen. Your fire base appears at the lower left hand corner.

The alien spaceship comes onto the screen from the top right hand corner and flies out into the middle. You must manoeuvre your fire base under the alien and press the fire key. The alien will then drop his word bomb on your city. You have to try and unscramble the word and type it in correctly before the bomb hits. When you have typed the word in correctly, it will fly up from the ground and stop the word bomb from falling any further.

All the word bombs are contained in the DATA statements in lines 12000 to 122000. Line 70 READs this DATA into an array and line 503 then picks a word out of this array at random. The routine in lines 505 to 527 scrambles the word and turns it into our word bomb.

Lines 20 to 60 set up the sprites registers and READ sprite DATA from the DATA store in lines 11900 to 11997.

```

5 REM *** UNSCRAMBLE ALIEN ***
10 GOSUB 6000
15 POKE 650,255:DIM S$(238),G$(7):W=3
20 V=53248:POKE 2040,192:POKE 2041,193:POKE
    2042,194:POKE 2043,195
30 FORS=12288T012350:READX:POKE S,X:NEXT:POKEV+21
    ,0
35 FORS=12352T012414:READX:POKE S,X:NEXT:POKEV+40
    ,7
37 FORS=12416T012478:READX:POKE S,X:NEXT
40 FORS=12480T012543:READX:POKE S,X:NEXT
50 POKEV+39,0:POKEV+29,1:POKEV+41,0:POKEV+5,140
    :POKEV+4,25:POKEV+23,12
60 POKEV+42,0
70 FORT=0T0236:READS$(T):NEXT
100 POKE53280,254:POKE53281,246
110 GOTO 140
120 PRINT"  " UNSCRAMBLE ALIEN
    "
125 REM CLR SCRN LT BLUE REV ON
130 RETURN
140 GOSUB 120
141 PRINT"  "
    "
142 PRINT"  "
    "
143 REM "T" IS UP CRSR " " IS REV ON " " IS
    REV OFF
145 PRINT"  "
    "
150 PRINT"  " " YOU MUST DEFEND YOUR CITY BY
    " "
151 REM WHITE
155 PRINT"  " "
    " "
160 PRINT"  " " TYPING IN THE WORD TO THE
    JUMBLED " "
161 REM WHITE
165 PRINT"  " " "
170 PRINT"  " " 'WORD BOMB' THAT THE ALIENS ARE

```

[illegible]

```

      KEY "
305 REM DOWN CRSR REV ON LT BLUE CRSR DOWN REV
      ON LT GREEN
310 GET A$:IF A$=""THEN 310
312 GOSUB 500
315 FU$="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
320 PRINT"UNJUMBLE ALIENS
      XXXX";GOSUB500000
321 D=INT(RND(1)*150)+30:P=24:POKEV+21,3:X=180:
      Y=60:POKEV+1,Y:POKEV+3,220
330 POKEV+0,X:POKEV+2,P
380 GET A$:POKE 650,255
390 IF A$="A" AND P-6>30 THEN P=P-6
400 IF A$="D" AND P+6<186 THEN P=P+6
401 IFA$="S" AND P>=D THEN 600
420 IF X-1<>D THEN X=X-1
430 IF X=D THEN 380
440 GOTO 330
500 G1=0:G2=0:G3=0:G4=0:G5=0
501 V=53248:POKE V+21,0
503 T=INT(RND(1)*237)+0:F$=S$(T)
505 LE=LEN(F$)
510 FOR G=1 TO LE:G$(G)=MID$(F$,G,1):NEXT
520 G1=INT(RND(1)*LE)+1
521 G2=INT(RND(1)*LE)+1:IF G2=G1 THEN 521
522 G3=INT(RND(1)*LE)+1:IF G3=G1 OR G3=G2
      THEN 522
524 G4=INT(RND(1)*LE)+1:IF G4=G1 OR G4=G2 OR
      G4=G3 THEN 524
525 IF LE<5 THEN G5=0
526 FOR T=1 TO LE:IF G1<>T AND G2<>T AND G3<>T
      AND G4<>T THEN G5=T
527 NEXT T:RETURN
528 TI$="0000000":LT=28:A$="":DR=6
530 N$=LEFT$(FU$,DR)
532 IF DR>6 THEN PRINT"UNJUMBLE ALIENS";N$;TAB(LO);G$(G1);G$(
      G2);G$(G3);G$(G4);G$(G5)
533 REM HOME WHITE
535 IF DR>7 THEN PRINT"UNJUMBLE ALIENS";LEFT$(FU$,DR-1);TAB
      (LO);" "
560 IFAN$=""THENPRINT"UNJUMBLE ALIENS";TAB(28);"

```

```

7":LT=28:REM HOME DOWN UP CURSR
561 GET A$
562 PRINT"*****"TAB(28)"LIVES:";W:PRINT"
*****"TAB(28)"WORDS"
563 PRINT"*****"TAB(28);":LEFT:"10-M
564 PRINT"    ":PRINT"*****"TAB(29);":YOU
TYPED:"
565 IF A$="" THEN 570
566 PRINT"    "
567 PRINT"*****"TAB(28);":YOU TYPED:"PRINTTAB
(LT);A$:AN$=AN$+A$:A$="":LT=LT+1
568 REM USE ? FOR PRINT OR LINE WON'T FIT
570 IF LT=28=LE GOTO 1000
580 DR=VAL(TI$)+6:IF DR=25 THEN 3000
590 GOTO 530
600 POKE 650,0:SH=0:FOR CH=36 TO 180 STEP 6
605 IF CH>36 THEN SH=SH+1
606 IF CH=P THEN LO=SH-2
607 IF P=36 OR P=42 THEN LO=0
610 NEXT:POKE V+16,12:POKE V+21,7
640 GOSUB20000:PRINT"*****"TAB(30);":
G$(G1);G$(G2);G$(G3);G$(G4);G$(G5)
645 REM HOME DOWN CURSR GREEN REV ON
650 PRINT"*****":FOR T=1 TO 14:PRINTTAB(28)"
":NEXT T
660 GOTO 528
1000 IF AN$=F$ THEN 2000
1010 PRINT"*****WRONG"AN$="":LT=28:GOTO 530
1020 REM HOME DOWN REV ON
2000 FOR KI=20 TO DR STEP-1:PRINT"*****"LEFT$
(FU$,KI);TAB(LO);AN$:REM LT GREEN
2001 PRINTLEFT$(FU$,KI+1);TAB(LO);":FORG=
0TO20:NEXTG,KI:POKEV+21,7
2004 FOR P=0 TO 24:POKE L+P,0:NEXT:POKE V+6,25:
POKE V+7,140
2005 POKE L,240:POKE H,33:POKE A,8:POKE L+22,
104:POKE L+23,1:POKE L+24,79
2006 FOR PN=1 TO 15:POKE L+4,129:POKE V+21,7:
FOR T=1 TO 250:NEXT:POKE L+4,128
2007 POKE V+21,11:FOR T=1 TO 30:NEXT:NEXT:POKE

```

```

L+24,0
2000 FOR U=1 TO 1000:NEXT
2005 A#="":M=M+1:DR=6:POKE V+21,0
2010 IF M<11 THEN GOSUB 500:GOTO 315
2020 GOTO 4000
2040 A#="":M=M-1:DR=6
3000 PRINT"SOOOOBAD LUCK IT WAS ";F#;"
      ":GOSUB30000:M=M-1:DR=6:POKEV+21,0
3005 FOR U=1 TO 1000:NEXT
3010 IF M<1 THEN GOTO 5000
3020 GOSUB 500:GOTO 315
4000 FOR TD=1 TO 1000:NEXT
4010 PRINT "3":POKE 53280,6:POKE 53281,3
4030 PRINT "WOWWWW" TAB(1) "4YOU HAVE SAVED
      THE CITY FROM THE ALIENS"
4040 IF M=3 THEN PRINT "0" TAB(3) "WITHOUT
      LOSING ANY OF YOUR 3 LIVES"
4050 IF M=2 THEN PRINT "0" TAB(14) "5EXCELLENT"
      :GOTO 4070
4060 PRINT "0" TAB(12) "YOU LOST";3-M;"LIVES"
4070 GOSUB 30000
4080 END
5000 FOR TD=1 TO 1000:PRINT "3":POKE 53280,0:
      POKE 53281,0
5010 PRINT "WOWWWW" TAB(5) "3YOU HAVE LOST
      ALL THREE LIVES"
5020 PRINT "0" TAB(6) "3AFTER GUESSING ONLY";M:
      "WORDS"
5030 GOSUB 30000
5040 END
6000 PRINT "3":POKE 53280,4:POKE 53281,3
6010 PRINT "WOWWWW":PRINT TAB(12);
6020 GOSUB 6110:PRINT "6SAMPLEBNURC LINEA":FOR
      TD=1 TO 500:NEXT
6030 GOSUB 6110:PRINT "USNLRMABCE AEILN":FOR
      TD=1 TO 500:NEXT
6040 GOSUB 6110:PRINT "UNSMRLCABE AELIN":FOR
      TD=1 TO 500:NEXT

```



```

6050 GOSUB 6110:PRINT "UNSCRAMBLE ALIEN":GOSUB
    '30000
6060 RETURN
6110 PRINT "~~~~~":PRINT TAB(12);:RETURN
10000 PRINT"~~~~~ HIT ANY KEY TO CONTINUE"
10010 GET A$:IF A$="" THEN 10010
10020 RETURN
11900 DATA0,0,0,0,0,0,14,0,224,2,0,128,3,255,
    128,15,255,192,30,56,240,62,124
11910 DATA252,126,56,254,255,255,255,255,255,
    255,255,255,255,126,0,254,62,254
11920 DATA248,31,255,240,15,255,224,2,0,128,2,0,
    128,14,0,224,10,0,160,14,0
11940 DATA0,0,16,0,0,56,0,0,124,0,0,124,0,0,124,
    0,0,254,0,16,254,16,57,255,56,69
11950 DATA255,76,131,255,130,131,255,130,131,
    255,130,131,255,130,131,255,130
11960 DATA131,255,130,131,255,130,131,255,130,
    131,255,130,131,255,130,131,255
11970 DATA130,130,254,0,0
11975 DATA56,0,0,124,0,0,230,0,0,254,4,0,124,15,
    0,56,56,0,126,224,0,255,128,0
11980 DATA254,0,0,255,255,0,254,4,0,254,0,0,0,0,
    0,254,0,0,124,0,0,124,0,0,124,0
11985 DATA0,124,0,0,56,0,0,56,0,0,56,0,0,0,0,0,
    0,0,0
11990 DATA56,0,0,124,0,0,230,0,0,254,0,0,254,0,
    0,124,0,0,120,2,0,255,255,128
11995 DATA255,255,129,252,2,0,252,0,0,252,0,0,
    252,0,0,0,0,0,252,0,0,248,0,0,112
11997 DATA0,0,112,0,0,126,0,0
12000 DATAMORE,ROME,HAIR,GIRL,FOOD,PHONE,FARM,
    SHOT,HOUSE,GRASS,DATE
12010 DATAYOUNG,THERE,THEIR,WERE,WHERE,FOUND,
    POND,HORSE,RADIO,DOOM,PAPER,NEWS
12020 DATADEATH,KILL,CLASS,BURN,HANDS,FEET,HEAD,
    BODY,GLASS,SHOE,FIRE,NIGHT,WORN
12030 DATAWORK,PLAY,DOOR,VASE,LUNCH,BREAD,BIRD,
    DRAW,ROOM,ZEBRA,SOME,LIGHT,WRONG

```

12040 DATARIGHT, NAIL, PLAIN, QUEEN, KING, LAST, LEMON
 , BLACK, BOX, XMAS, QUAY, LIGHT
 12050 DATAMAID, MADE, RAISE, SNOOP, PEEK, POKE, NOSE,
 PRIZE, EDIT, LINE, BACK, RALLY
 12060 DATAROOT, SCREW, HUMP, POEM, VERSE, SONG, BUNK,
 EPIC, TIED, TILT, LILT, WILT
 12070 DATATORCH, MATCH, FIRE, SMOKE, STAND, MATHS,
 LEGAL, ROYAL, TURN, AMPLE, BOOT, SONIC
 12080 DATARIFLE, SPRAY, TABLE, GIRL, LAZY, SHIRT,
 SHORT, FLOOR, CLOTH, CARD, HEART
 12090 DATASUNNY, CLOUD, QUICK, PRIME, FOLD, HOLD, SOLD
 , COLD, MOULD, BOLD, FUNNY, BUNNY
 12100 DATAWOOD, GOOD, HOOD, QUAD, BRAY, DAZE, PHASE,
 GAZE, HAZE, CRAZE, MAZE, PRAY
 12110 DATARAZOR, TRAY, QUASH, QUILL, MILL, DRILL, SILL
 , KILL, PLUM
 12120 DATANAME, QUIET, MUTE, NOISY, ROVER, RANK, SLATE
 , GREAT, LARGE, ZERO, REAL, VAGUE
 12130 DATALIKE, ABLE, BEEN, BEAN, SHAFT, ARROW, CRAG,
 BLEAK, LATE, FATE, GRATE, ERECT
 12140 DATAMUFF, ALERT, ALOOF, SPEED, ABOUT, TARDY,
 DETER, OVER, STORE, PROXY, PRUDE
 12170 DATABEAT, PULSE, LACK, NEED, WANT, FUND, DRAB,
 FLUSH, SCARE, ORATE, SERF, SLAVE
 12180 DATADIRECT, QUOTE, SCAN, PLACE, GLUE, STICK,
 VEST, AGREE, FADE, CLUB, TRIBE, CROWD
 12190 DATASHAKE, CALM, BEGIN, FORTH, TRICK, PUNCH,
 CHEAT, ROOK, GULL, FRAUD, SWAY, WAVE
 12200 DATASWIRL, WHIRL, SNAG, CHOP, SAUCE, WIDE, ALLY,
 COUNT, SOLVE, SHAME, FORCE, SCOPE
 12210 DATA, ONLY, BUSH, PHONE, BROWN, BLUE, WHITE,
 NAVY, ARMY, PINK, GREEN, APPLE, PEAR
 20000 HK=54296:K=54276:A=54277:H=54273:L=54272
 20005 FORT=0T024:POKEL+T,0:NEXT
 20010 FORX=15T00STEP-1:POKEHK,X:POKEK,129:POKEA,
 15:POKEH,40:POKEL,200:NEXT
 20020 POKEK,0:POKEA,0:POKEHK,0:RETURN
 30000 S=54272:FORT=0T024:POKES+T,0:NEXT

```

30010 POKES+1,130:POKES+5,9:POKES+15,30:POKES+24
,15
30020 FORL=0TO5:POKES+4,21:FORT=0TO200:NEXTT:
POKES+4,0:FORT=0TO200:NEXTT,L
30030 RETURN
50000 PRINT"3":G=0
50005 PRINT"XXXXXXXXXXXXXXXXXXXXTAB(G);"      "
50006 REM ORANGE HOME DOWN CRSR
50010 PRINTTAB(G);"      /  ^ "
50020 PRINTTAB(G);" /  ^  | "
50030 PRINTTAB(G);" | | | L | "
50040 PRINTTAB(G);" | | | /  | "
50050 PRINTTAB(G);" | | | | | | "
50060 PRINTTAB(G);" | | | | | | | "
50070 PRINTTAB(G);" | | | | | | | "
50080 PRINTTAB(G);" | | | | | | | "
50090 IF G=26 THEN 50110
50100 G=G+13:GOTO 50005
50110 RETURN

```

REVERSI

You can now challenge your computer to a game of Reversi. The computerised version of this popular board game (often called Othello) should give you many hours of hard work as you try to defeat the mighty 64. You will find that the computer plays this game very well and you'll be hard-pressed to win.

Reversi is played on an eight by eight board and players take it in turns to place a piece on the board. The pieces are put onto the board in such a way that one or more of your opponent's pieces are *trapped* between two of your pieces. These trapped pieces then become yours. At the end of the game, the player with the most pieces wins.

The game starts with each player already having two pieces on the board. Your playing pieces are the yellow circles and the computer's pieces are the blue diamonds.

To make the game a little more interesting a time routine has been added. This gives both you and the computer a set amount of time in which to make your move. If either player fails to make a move before the time runs out, that player forfeits his turn.

At the beginning of the game you will be asked if you wish to have the first move. After you have answered this question, the screen will clear and the board will be printed up.

To the right of the board is a box which indicates that a move is being considered and, beneath this is the time count down display which shows how much time is left for a move.

The computer will prompt you when it is your turn to move. First enter the number down the side of the board and then the number across the board to indicate which square you wish to place your piece in. If you cannot make a move, enter a zero. The prompt routine is located in lines 80 to 150.

The board is printed by lines 700 to 920. This routine prints the grid, checks to see which locations contain playing pieces and to which player they belong. It also checks to make sure that both players are still in the game. Under the playing board it prints how many pieces each player has, and who has the current turn.

169

```

270 REM
280 PRINT"D"
290 GOSUB 1110:GOSUB 700
300 IFA$="1"THEN TI$="000045":GOTO550
310 PRINT TAB(9)"MY MOVE  ":TI$="000000"
320 FLAG=U1:GOTO 160
330 S=0:T=X:H=0:FOR A=2 TO 9:FOR B=2 TO 9:IF A
    (A,B)<>46 THEN 500
340 Q=0:FOR C=-1 TO 1:FOR D=-1 TO 1:K=0:F=A:G=B
350 IF A(F+C,G+D)<>S THEN 370
360 K=K+U1:F=F+C:G=G+D:GOTO 350
370 IF A(F+C,G+D)<>T THEN 390
380 Q=Q+K
390 NEXT:NEXT
400 IF A=A2 OR A=9 THEN Q=Q*A2
410 IF B=A2 OR B=9 THEN Q=Q*A2
420 IF A=3 OR A=8 THEN Q=Q/A2
430 IF B=3 OR B=8 THEN Q=Q/2
440 IF (A=A2 OR A=9) AND (B=3 OR B=8)THEN Q=Q/A2
450 IF (A=3 OR A=8) AND (Z=2 OR Z=9) THEN Q=Q/A2
460 IF Q<H OR Q=0 OR (RND(U1)>P3 AND Q=H) THEN
    500
470 H=Q:M=A:N=B
480 GOSUB 220
490 FLAG=2:GOTO 160
500 FLAG=3:GOTO 160
510 GOSUB 220:NEXT:NEXT:IF H=0 AND R=0 THEN 1000
520 IF H=0 THEN 540
530 GOSUB 930
540 GOSUB 700
550 J=VAL(TI$):S=X:T=0:PRINT TAB(9) "YOUR GO ";
560 IF VAL(TI$) >42 THEN J=42
570 C$="":TI$="000000"
580 GET B$:ON LEN(C$)+1GOSUB 110,80
590 GOSUB 220
600 FLAG=4:GOTO 160
610 IF B$="" THEN 580
620 IF B$="-" THEN C$="":GOTO 580
630 IF B$="0" THEN R=0:GOTO 690
640 IF ASC(B$)<48 OR ASC(B$)>57 THEN 580
650 GOSUB 140:PRINT B$:C$=C$+B$:IF LEN(C$)=2
    THEN R=VAL(C$):GOTO 670

```

```

660 GOTO 580
670 M=INT(R/10)+1:N=R-10*INT(R/10)+1
680 GOSUB 930
690 GOSUB 700:GOTO310
700 PRINT "M":C=0:H=0
710 PRINT:PRINT TAB(12)"REVERSI":PRINT:REM
    ORANGE
720 PRINT TAB(11)"12345678":REM WHITE
730 PRINT
740 FOR B=2 TO 9:PRINT TAB(8)B-1:
750 FOR D=2 TO 9:IF A(B,D)=X THEN C=C+1:PRINT
    "O+":GOTO 770:REM LT BLUE
760 IF A(B,D)=0THEN PRINT" ":H=H+1:REM YELLOW
770 PRINT" ":REM WHITE
780 IF A(B,D)<>X AND A(B,D)<>0 THEN PRINT".":
790 NEXT
800 PRINT B-1:NEXT
810 PRINT:PRINT TAB(11)"12345678"
820 PRINT
830 IF C+H=64 THEN 1000
840 IF C=0 THEN 1000
850 IF H=0 THEN 1000
860 PRINT TAB(9)"I HAVE: ";
870 IF C>10 THEN PRINT SPC(1):C:PRINT
880 IF C<10 THEN PRINT SPC(2):C:PRINT
890 PRINT TAB(9)"YOU HAVE: ":REM GREEN WHITE
900 IF H>10 THEN PRINT SPC(1):H
910 IF H<10 THEN PRINT SPC(2):H
920 PRINT:RETURN
930 FOR C=-1 TO1:FOR D=-U1 TO U1:F=M:G=N
940 IF A(F+D,G+D)<>S THEN 960
950 F=F+D:G=G+D:GOTO 940
960 IF A(F+D,G+D)<>T THEN 990
970 A(F,G)=T:IF M=F AND N=G THEN 990
980 F=F-D:G=G-D:GOTO 970
990 NEXT:NEXT:RETURN
1000 REM
1010 FOR I=1 TO 1000:NEXT
1020 PRINT "*****"
1030 IF C>H THEN PRINT SPC(24);"I WON":C:H
1040 IF C<H THEN PRINT SPC(24);"YOU WON":H:C
1050 IF H=C THEN PRINT SPC(24);"IT'S A DRAW !!!"
1060 PRINT:PRINT SPC(24);"ANOTHER (Y=YES, N=NO)"

```

172

CHESSBOARD NIM

This program plays an interesting variation on the game of Nim. The idea of the game is to start with a random number of playing pieces, matches are often used, then two players take it in turns removing a number of pieces from the pile. The player who is forced to pick up the last piece loses.

The computer picks a random number of playing pieces and places them on a chessboard. It then randomly sets a limit to the number of pieces which can be removed at any one turn. This is done in line 60.

The routine from line 410 to 480 places the number of playing pieces into the A array and then lines 90 to 110 print out the relevant information.

The board and playing pieces are printed out in the routine in lines 120 to 210. You can change the shape of the playing pieces by altering the value in line 420.

```
10 REM CHESSBOARD NIM
15 POKE 53280,6:POKE 53281,11
30 S=0:C=0
50 Z=INT(11*RND(1))+21
60 F=INT(3*RND(1))+3
70 DIM A(32)
80 GOTO 410
90 IF S=0 THEN PRINT TAB(7) "♙PIECES ON THE
    BOARD: ";Z:REM GREEN
100 IF S=0 THEN PRINT "♚THE MAXIMUM NUMBER YOU
    CAN REMOVE IS:";F:REM LT RED
110 IF C>0 AND S=0 THEN PRINT TAB(7) "♜YOU TOOK"
    ;C;"AND I TOOK";D:REM YELLOW
120 FOR A=0 TO 3
130 PRINT
140 FOR B=0 TO 3
150 PRINT "♜" TAB(15)CHR$(A(29+A-B*8));CHR$(166)
    ;:REM PURPLE
```

```

160 NEXT B
170 PRINT
180 FOR B=0 TO 3
190 PRINT TAB(15)CHR$(166);CHR$(A(25+A-B*8));
200 NEXT B
210 NEXT A
215 PRINT
220 IF S=1 THEN PRINT TAB(16) "YOU WIN":END
    :REM LT BLUE
230 IF S=2 THEN PRINT TAB(17)"I WIN":END:
    REM LT BLUE
250 PRINT
260 PRINT TAB(7) "HOW MANY WILL YOU TAKE":
    REM ORANGE
270 INPUT C
280 IF C<1 OR C>F THEN GOTO 270
290 Z=Z-C
300 IF Z>0 THEN 330
310 S=2
320 IF S=2 THEN GOTO 410
330 D=Z-1-INT((Z-1)/(F+1))*(F+1)
340 IF D=0 AND NOT Z=1 THEN D=INT(F*RND(1))+1
350 IF NOT D<Z THEN 340
360 IF Z<F+2 AND INT(5*RND(1))+1=5 THEN D=D+INT
    (2*RND(1))+1-INT(2*RND(1))+1
370 IF D>F THEN GOTO 330
375 IF D>Z THEN 330
380 IF D<0 THEN D=0
390 Z=Z-D
400 IF Z=0 THEN S=1
410 FOR A=1 TO Z
420 A(A)=113
430 NEXT A
440 FOR A=Z+1 TO 32
450 A(A)=32
460 NEXT A
470 PRINT " ":REM CLEAR SCREEN
480 GOTO 90

```

APPENDIX

Writing your own games

WRITING YOUR OWN GAMES

To write an exciting game of your own you must first start with a specific idea. This may sound a bit obvious but unless you begin with a definite goal in mind you will find that it isn't easy to produce a game which will provide a challenge and still be fun to play.

Games fall into three basic categories:-

- Board games, such as *Checkers*, *Chess* and *Solitaire*.
- Arcade games, such as *Pacman* and *Space Invaders*.
- Games of chance, such as card games and dice games.

You can find ideas for your own games by watching arcade games, looking in toy shops at the large variety of games available and by reading books about the various types of games of chance. However you must be very careful not to infringe the copyright on any commercial products. Even if you start out to copy an established arcade game, you will probably find that at some point during the development of the game it will begin to change and will probably finish up bearing little resemblance to the original. Computer programs tend to develop a life of their own as they expand!

Before you start programming it is essential, as we pointed out earlier, that you have a firm idea of what you need to do to achieve the final product. Set out, on a piece of paper, all the phases of the game, all the functions the computer will need to perform and what actions are required from the human player. Does the computer have to play an intelligent game, as in *Chess* or does it have to provide a system of random events such as *Roulette*?

If you are writing a simulation such as *Sunnybrook Farm*, you will need to think out in advance all the factors that need to be taken into consideration and how they will interact. In a game such as this it is advisable to make a list of all the variable names you are going to use and what they stand for. This will prevent problems arising if you happen to use the same variable name twice in different routines. The first working version of *Battle of Britain* suffered from this problem. Roger used the variable H1 to keep

track of the number of Heinkel He111 bombers. Robert had used the same variable name, H1, as the value of the high frequency sound in one of the sound routines. As a result the numbers of He111 bombers rose and fell alarmingly during the course of the game!

Once you have the plan of your program, think out in advance any programming techniques that will be required. If it is an arcade type game, will sprites be needed or will programmable characters do the job better? Will arrays need to be set up? If they are, can they be filled with random elements or will known data be read in to fill the array?

When all these factors are clear in your mind, block out the program on paper, putting in all the routines which will be needed in an order which will make for efficient running and an easy to read, orderly program. You will see that the larger programs in this book such as *Sunnybrook Farm*, *Roman Empire* and *Battle of Britain* have been written using a control program in the first few lines which then calls up a series of subroutines as they are required. This is a very efficient method of programming, which is often called 'module programming'. It keeps the program neat and tidy with all related functions together in the one routine. This makes reading the program or debugging much easier. For instance, in the case of *Roman Empire*, if the information screen is not performing correctly, you know that the problem must be between lines 5800 and 5950, the routine which is called up to print the Senate Report.

Put the routines which are only used once, such as the initialisation routine (in which you set up variables, sound generators, sprites, etc) and the title screen routine right at the end of the program where they are out of the way. Then try to keep routines in the order which they are most required. If you are using one routine regularly and, you place it right at the end of a long program, the computer has to run through nearly every line number in the program looking for your routine each time you call for it.

In moving graphics or arcade type games, the speed of the game is the most important thing. If a game is too fast it can be slowed down using time delay loops, but is very hard to speed up a game which runs too slowly. In games such as *Descent Into Chaos* and *Space Race*, we have used a core program which only contains the basic elements required to make the game run. All other

routines are outside this loop and are called up only as they are needed. This core program should be kept as clean as possible for fast, efficient running.

If you have a program which requires input from the human player, make sure that you have used sufficient error trapping routines to prevent *all* illegal inputs being accepted by the computer. You should also make sure that any information the player needs stays on the screen long enough for him or her to read it!

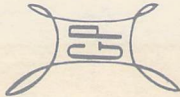
You must allow for as many unlikely events as you possibly can in your game design. If you have developed the game entirely by yourself, you are then the world's greatest (and only) expert on how your game should be played. You can't assume that everyone else who plays your game is going to play intelligently or obey your rules.

One of the first things we do when a program is submitted to us for publication is to play the most outrageously bad game we possibly can. For instance, in an Adventure program we will try to walk through walls, pick up objects that don't exist and any other similar thing we can think of. In moving graphics games, we will try to move off screen, shoot friendly objects and generally try to make the program break down. In a board game we will move the wrong way and make as many illegal or stupid moves as possible to see how the program copes with problems.

To be a winner, your game must be interesting to play and provide a challenge for the player so that you hold his or her interest. Lots of sound and colour as well as good routines to cover failure and success will add a lot to any game.

Finally, your game must be thoroughly play-tested. You must make sure that every routine, no matter how infrequently it occurs, works every time under all circumstances.

If you have written a game which you think is an interesting, exciting and challenging program, why not submit it for publication and let others enjoy your programming skills. We are always interested in hearing from new programmers and can be contacted via the publisher.



GARCIA PEÑA

Librería · Distribución

Director Propietario

Manuel García Peña

D. N. I. 560.612

Cavanilles, 52 - Teléfs. 252 44 02 - 03

MADRID - 7

SR. D.

JORGE MONFERRER MONTOLIU

DOCTOR BARRAGUER, 12

08033-CASTELLDEFELS.

BARCELONA.

MADRID, 14 DE FEBRERO DE 19 85 CUENTA N.º 00000000 FACTURA N.º 00004640

REFERENCIA DE SU PEDIDO: *23.12.84		ENVIO POR:		FORMA DE PAGO: REEMBOLSO	
N.º ALBARAN	CODIGO LIBRO	CANTIDAD	AUTOR · TITULO		
	00000000	1	GAMES COMMODORE 64 COMPUTER PLAY		
IMPORTE BRUTO 2.650,00		DESCUENTO		PRECIO 2.650	IMPORTE PESETAS 2.650,00
GTOS. ENVIO		IMPORTE NETO FACTURA ****2.650,00			

PRECIOS APLICADOS DE ACUERDO CON LA LEGISLACION VIGENTE.

GAMES

COMMODORE 64 COMPUTERS PLAY

If you are the owner of a Commodore 64 microcomputer, then bring a new dimension of fun into your life with this book of exciting games and adventures for your machine.

Once seated in front of your microcomputer you will find yourself rapidly transposed to the cockpit of a Spitfire as you relive the Battle of Britain. Time is no object as you travel back to take the throne at the head of the vast Roman Empire. But if you find this pace a little too hot then settle down to challenge the might of the Commodore 64 directly in a game of Reversi.

Other challenges include:

- ★ testing your reflexes in a Descent into Chaos
- ★ skiing down difficult mountain slopes
- ★ braving the dangers of a Cosmic Journey
- ★ attempting to stay alive in Survival
- ★ the responsibility of running Sunnybrook Farm
- ★ using your imagination in visualizing 3-D Noughts and crosses

0 201 14428 X



Addison-Wesley Publishing Company

GB £ NET IMPORTADOR N.º 389
GARCIA PEÑA
ISBN 0-201-14428-4
Pts. 11.000

9 780201 144284